ITRS-Log-Analytics-7.x Documentation

Release 7.0.2

Aug 14, 2020

Contents

About

ITRS Log Analytics User Guide

Software ver. 7.x

Introduction

ITRS Log Analytics is innovation solution allowing for centralize IT systems events. It allows for an immediately review, analyze and reporting of system logs - the amount of data does not matter. ITRS Log Analytics is a response to the huge demand for storage and analysis of the large amounts of data from IT systems. ITRS Log Analytics is innovation solution that responds to the need of effectively processing large amounts of data coming from IT environments of today's organizations. Based on the open-source project Elasticsearch valued on the marked, we have created an efficient solution with powerful data storage and searching capabilities. The System has been enriched of functionality that ensures the security of stored information, verification of users, data correlation and visualization, alerting and reporting.
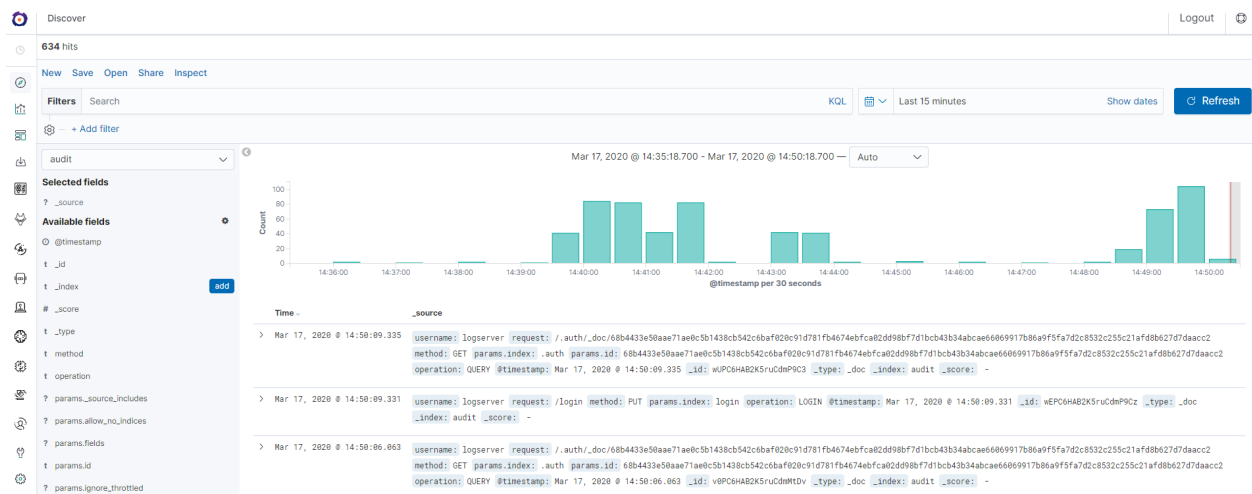
ITRS Log Analytics project was created to centralize events of all IT areas in the organization. We focused on creating a tool that functionality is most expected by IT departments. Because an effective licensing model has been applied, the solution can be implemented in the scope expected by the customer even with very large volume of data. At the same time, the innovation architecture allows for servicing a large portion of data, which cannot be dedicated to solution with limited scalability.

2.1 Elasticsearch

Elasticsearch is a NoSQL database solution that is the heart of our system. Text information send to the system, application and system logs are processed by Logstash filters and directed to Elasticsearch. This storage environment creates, based on the received data, their respective layout in a binary form, called a data index. The Index is kept on Elasticsearch nodes, implementing the appropriate assumptions from the configuration, such as:

- Replication index between nodes,

- Distribution index between nodes.

The Elasticsearch environment consists of nodes:

- Data node - responsible for storing documents in indexes,

- Master node - responsible for the supervisions of nodes,

- Client node - responsible for cooperation with the client.

Data, Master and Client elements are found even in the smallest Elasticsearch installations, therefore often the environment is referred to as a cluster, regardless of the number of nodes configured. Within the cluster, Elasticsearch decides which data portions are held on a specific node.

Index layout, their name, set of fields is arbitrary and depends on the form of system usage. It is common practice to put data of a similar nature to the same type of index that has a permanent first part of the name. The second part of the name often remains the date the index was created, which in practice means that the new index is created every day. This practice, however, is conventional and every index can have its own rotation convention, name convention, construction scheme and its own set of other features. As a result of passing document through the Logstash engine, each entry receive a data field, which allow to work witch data in relations to time.

The Indexes are built with elementary part called shards. It is good practice to create Indexes with the number of shards that is the multiple of the Elasticsearch data nodes number. Elasticsearch in 7.x version has a new feature called Sequence IDs that guarantee more successful and efficient shard recovery.

Elasticsearch use the *mapping* to describes the fields or properties that documents of that type may have. Elasticsearch in 7.x version restrict indices to a single type.# Kibana #

2.2 Kibana

Kibana lets you visualize your Elasticsearch data and navigate the Elastic Stack. Kibana gives you the freedom to select the way you give shape to your data. And you don't always have to know what you're looking for. Kibana core ships with the classics: histograms, line graphs, pie charts, sunbursts, and more. Plus, you can use Vega grammar to design your own visualizations. All leverage the full aggregation capabilities of Elasticsearch. Perform advanced time series analysis on your Elasticsearch data with our curated time series UIs. Describe queries, transformations, and visualizations with powerful, easy-to-learn expressions. Kibana 7.x has two new feature - a new "Full-screen" mode to viewing dashboards, and new the "Dashboard-only" mode which enables administrators to share dashboards safely.# Logstash #

2.3 Logstash

Logstash is an open source data collection engine with real-time pipelining capabilities. Logstash can dynamically unify data from disparate sources and normalize the data into destinations of your choice. Cleanse and democratize all your data for diverse advanced downstream analytics and visualization use cases.

While Logstash originally drove innovation in log collection, its capabilities extend well beyond that use case. Any type of event can be enriched and transformed with a broad array of input, filter, and output plugins, with many native codecs further simplifying the ingestion process. Logstash accelerates your insights by harnessing a greater volume and variety of data.

Logstash 7.x version supports native support for multiple pipelines. These pipelines are defined in a *pipelines.yml* file which is loaded by default. Users will be able to manage multiple pipelines within Kibana. This solution uses Elasticsearch to store pipeline configurations and allows for on-the-fly reconfiguration of Logstash pipelines.# ELK #

2.4 ELK

"ELK" is the acronym for three open source projects: Elasticsearch, Logstash, and Kibana. Elasticsearch is a search and analytics engine. Logstash is a server-side data processing pipeline that ingests data from multiple sources simultaneously, transforms it, and then sends it to a "stash" like Elasticsearch. Kibana lets users visualize data with charts and graphs in Elasticsearch. The Elastic Stack is the next evolution of the ELK Stack.

Data source and application management

3.1 Data source

Where does the data come from?

ITRS Log Analytics is a solution allowing effective data processing from the IT environment that exists in the organization.

The Elsasticsearch engine allows building a database in witch large amounts of data are stored in ordered indexes. The Logstash module is responsible for load data into Indexes, whose function is to collect data on specific tcp/udp ports, filter them, normalize them and place them in the appropriate index. Additional plugins, that we can use in Logstash reinforce the work of the module, increase its efficiency, enabling the module to quick interpret data and parse it.

Below is an example of several of the many available Logstash plugins:

exec - receive output of the shell function as an event;

imap - read email from IMAP servers;

jdbc - create events based on JDC data;

jms - create events from Jms broker;

Both Elasticsearch and Logstash are free Open-Source solutions.

More information about Elasticsearch module can be find at: https://github.com/elastic/elasticsearch

List of available Logstash plugins: https://github.com/elastic/logstash-docs/tree/master/docs/plugins

3.2 System services

For proper operation ITRS Log Analytics requires starting the following system services:

- elasticsearch.service - we can run it with a command:

```
    systemctl start elasticsearch.service
```

we can check its status with a command:

systemctl status elasticsearch.service

```
[root@collector1 centos]# systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled; vendor preset: disabled)
   Active: active (running) since Fri 2020-03-20 13:03:21 UTC; 4 days ago
     Docs: http://www.elastic.co
 Main PID: 1586 (java)
   CGroup: /system.slice/elasticsearch.service
           └─1586 /bin/java -Xms4g -Xmx4g -Djava.security.manager -Djava.security.policy=/usr/share/elasticsearch/plugins/elasticsearch-auth/java.poli...
```

- kibana.service - we can run it with a command:

```
    systemctl start kibana.service
```

we can check its status with a command:

systemctl status kibana.service

```
[root@collector1 centos]# systemctl status kibana
● kibana.service - Kibana
   Loaded: loaded (/etc/systemd/system/kibana.service; enabled; vendor preset: disabled)
   Active: active (running) since Fri 2020-03-20 14:08:37 UTC; 4 days ago
 Main PID: 17248 (node)
   CGroup: /system.slice/kibana.service
           └─17248 /usr/share/kibana/bin/../node/bin/node --no-warnings /usr/share/kibana/bin/../src/cli -c /etc/kibana/kibana.yml

Mar 24 14:40:39 collector1 kibana[17248]: {"type":"response","@timestamp":"2020-03-24T14:40:39Z","tags":[],"pid":17248,"method":"get","status...pplicatio
Mar 24 14:40:39 collector1 kibana[17248]: Radius selection : undefined
Mar 24 14:40:39 collector1 kibana[17248]: Token :
Mar 24 14:40:39 collector1 kibana[17248]: Username : undefined
Mar 24 14:40:39 collector1 kibana[17248]: {"type":"response","@timestamp":"2020-03-24T14:40:39Z","tags":[],"pid":17248,"method":"get","status...x-csrf-to
Mar 24 23:00:00 collector1 kibana[17248]: PDF Export tasks in index :  0
Mar 24 23:00:00 collector1 kibana[17248]: No Tasks in taskmanagemnt index for export type dashboard
Mar 24 23:00:00 collector1 kibana[17248]: CSV Export tasks in index :  0
Mar 24 23:00:00 collector1 kibana[17248]: No Tasks in taskmanagemnt index for export type csv
Mar 25 00:00:02 collector1 kibana[17248]: {"type":"log","@timestamp":"2020-03-25T00:00:02Z","tags":["\u001b[34mwazuh\u001b[39m","monitoring",... index."}
Hint: Some lines were ellipsized, use -l to show in full.
```

- logstash.service - we can run it with a command:

```
    systemctl start logstash.service
```

we can check its status with a command:

systemctl status logstash.service

```
[root@collector1 centos]# systemctl status logstash
● logstash.service - logstash
   Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2020-03-24 08:12:22 UTC; 1 day 3h ago
 Main PID: 16987 (java)
   CGroup: /system.slice/logstash.service
           └─16987 /bin/java -Xms1g -Xmx1g -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMSInitiatingOccupancyOnly -Djava.awt...

Mar 24 08:13:15 collector1 logstash[16987]: [2020-03-24T08:13:15,642][INFO ][logstash.inputs.udp      ] UDP listener started {:address=>"0.0.0...>"2000"}
Mar 24 08:13:15 collector1 logstash[16987]: [2020-03-24T08:13:15,689][WARN ][logstash.outputs.elasticsearch] Restored connection to ES instanc...:9200/"}
Mar 24 08:13:15 collector1 logstash[16987]: [2020-03-24T08:13:15,715][INFO ][logstash.outputs.elasticsearch] New Elasticsearch output {:class=...:9200"]}
Mar 24 08:13:15 collector1 logstash[16987]: [2020-03-24T08:13:15,741][INFO ][logstash.outputs.elasticsearch] Using default mapping template
Mar 24 08:13:15 collector1 logstash[16987]: [2020-03-24T08:13:15,743][INFO ][logstash.outputs.elasticsearch] Attempting to install template {:...."=>"mess
Mar 24 08:13:15 collector1 logstash[16987]: [2020-03-24T08:13:15,754][INFO ][logstash.filters.geoip   ] Using geoip database {:path=>"/usr/sha...y.mmdb"}
Mar 24 08:13:15 collector1 logstash[16987]: [2020-03-24T08:13:15,890][INFO ][logstash.inputs.file     ] No sincedb_path set, generating one ba....json"}
Mar 24 08:13:15 collector1 logstash[16987]: [2020-03-24T08:13:15,945][INFO ][filewatch.observingtail  ] START, creating Discoverer, Watch with...lections
Mar 24 08:13:16 collector1 logstash[16987]: [2020-03-24T08:13:16,010][INFO ][logstash.pipeline        ] Pipeline started successfully {:pipeli...7 run>"}
Mar 24 08:13:18 collector1 logstash[16987]: [2020-03-24T08:13:18,370][INFO ][logstash.agent           ] Successfully started Logstash API endp...t=>9600}
Hint: Some lines were ellipsized, use -l to show in full.
```

3.3 First configuration steps

3.3.1 Run the instalation

The ITRS Log Analytics installer is delivered as:

- RPM package itrs-log-analytics-data-node and itrs-log-analytics-client-node

- "install.sh" installation script

Installation using the RPM package

1. Install OpenJDK / Oracle JDK version 11:

```
yum -y -q install java-11-openjdk-headless.x86_64
```

2. Select default command for OpenJDK /Oracle JDK:

```
alternatives --config java
```

3. Upload Package

```
scp ./itrs-log-analytics-data-node-7.0.1-1.el7.x86_64.rpm root@hostname:~/
scp ./itrs-log-analytics-client-node-7.0.1-1.el7.x86_64.rpm root@hostname:~/
```

4. Install ITRS Log Analytics Data Node

```
yum install ./itrs-log-analytics-data-node-7.0.1-1.el7.x86_64.rpm
```

5. Verification of Configuration Files

   Please, verify your Elasticsearch configuration and JVM configuration in files:

   - /etc/elasticsearch/jvm.options – check JVM HEAP settings and another parameters

```
## -Xms4g
## -Xmx4g
# Xms represents the initial size of total heap space
# Xmx represents the maximum size of total heap space
-Xms600m
-Xmx600m
```

   - /etc/elasticsearch/elasticsearch.yml – verify elasticsearch configuration file

6. Start and enable Elasticsearch service If everything went correctly, we will start the Elasticsearch instance:

```
systemctl start elasticsearch
```

```
systemctl status elasticsearch
 elasticsearch.service - Elasticsearch
   Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled; vendor
→preset: disabled)
   Active: active (running) since Wed 2020-03-18 16:50:15 CET; 57s ago
     Docs: http://www.elastic.co
 Main PID: 17195 (java)
   CGroup: /system.slice/elasticsearch.service
           └─17195 /etc/alternatives/jre/bin/java -Xms512m -Xmx512m -Djava.
→security.manager -Djava.security.policy=/usr/share/elasticsearch/plugins/
→elasticsearch_auth/plugin-securi...

Mar 18 16:50:15 migration-01 systemd[1]: Started Elasticsearch.
Mar 18 16:50:25 migration-01 elasticsearch[17195]: SSL not activated for http and/
→or transport.
Mar 18 16:50:33 migration-01 elasticsearch[17195]: SLF4J: Failed to load class
→"org.slf4j.impl.StaticLoggerBinder".
```

(continues on next page)

```
Mar 18 16:50:33 migration-01 elasticsearch[17195]: SLF4J: Defaulting to no-
↪operation (NOP) logger implementation
Mar 18 16:50:33 migration-01 elasticsearch[17195]: SLF4J: See http://www.slf4j.
↪org/codes.html#StaticLoggerBinder for further details.
```

7. Check cluster/indices status and Elasticsearch version

   Invoke curl command to check the status of Elasticsearch:

```
curl -s -u $CREDENTIAL localhost:9200/_cluster/health?pretty
{
  "cluster_name" : "elasticsearch",
  "status" : "green",
  "timed_out" : false,
  "number_of_nodes" : 1,
  "number_of_data_nodes" : 1,
  "active_primary_shards" : 25,
  "active_shards" : 25,
  "relocating_shards" : 0,
  "initializing_shards" : 0,
  "unassigned_shards" : 0,
  "delayed_unassigned_shards" : 0,
  "number_of_pending_tasks" : 0,
  "number_of_in_flight_fetch" : 0,
  "task_max_waiting_in_queue_millis" : 0,
  "active_shards_percent_as_number" : 100.0
}
```

```
curl -s -u $CREDENTIAL localhost:9200
{
  "name" : "node-1",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "igrASEDRRamyQgy-zJRSfg",
  "version" : {
    "number" : "7.3.2",
    "build_flavor" : "oss",
    "build_type" : "rpm",
    "build_hash" : "1c1faf1",
    "build_date" : "2019-09-06T14:40:30.409026Z",
    "build_snapshot" : false,
    "lucene_version" : "8.1.0",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

   If everything went correctly, we should see 100% allocated shards in cluster health.

8. Install ITRS Log Analytics Client Node

```
yum install ./itrs-log-analytics-client-node-7.0.1-1.el7.x86_64.rpm
```

9. Start ITRS Log Analytics GUI

   Add service:

   • Kibana

- Cerebro

- Alert

to autostart and add port (5602/TCP) for Cerebro. Run them and check status:

```
firewall-cmd -permanent -add-port 5602/tcp
firewall-cmd -reload
```

```
systemctl enable kibana cerebro alert
```

```
systemctl start kibana cerebro alert
systemctl status kibana cerebro alert
```

Installation using "install.sh" scritp

During installation you will be ask about following tasks:

- install & configure Logstash with custom ITRS Log Analytics Configuration - like Beats, Syslog, Blacklist, Netflow, Wazuh, Winrm, Logtrail, OP5, etc;

- install the ITRS Log Analytics Client Node, as well as the other client-node dependencies;

- install the ITRS Log Analytics Data Node, as well as the other data-node dependencies;

- load the ITRS Log Analytics custom dashboards, alerts and configs;

Post installation steps

- copy license files to Elasticsearch directory

```
cp es.* /ust/share/elasticsearch/bin/
```

- configure Elasticsearch cluster settings

```
vi /etc/elaticserach/elasticsearch.yml

add all IPs of Elasticsearch node in the following directive:
discovery.zen.ping.unicast.hosts: [ "172.10.0.1:9300", "172.10.0.2:9300" ]
```

- start Elasticsearch service

```
systemc        start elasticsearch
```

- start Logstash service

```
systemctl start logstash
```

- start Cerebro service

```
systemctl start cerebro
```

- start Kibana service

```
systemctl start kibana
```

- start Alert service

```
systemctl start alert
```

- start Skimmer service

```
systemctl start skimmer
```

- Example agent configuration files and additional documentation can be found in the Agents directory:

    - filebeat

    - winlogbeat

    - op5 naemon logs

    - op5 perf_data

- For blacklist creation, you can use crontab or kibana scheduler, but the most preferable method is logstash input. Instructions to set it up can be found at `logstash/lists/README.md`

- It is recomended to make small backup of system indices - copy "small_backup.sh" script from Agents directory to desired location, and change `backupPath=` to desired location. Then set up a crontab:

```
0 1 * * * /path/to/script/small_backup.sh
```

- Redirect Kibana port 5601/TCP to 443/TCP

```
firewall-cmd --zone=public --add-masquerade --permanent
firewall-cmd --zone=public --add-forward-port=port=443:proto=tcp:toport=5601 --
↪permanent
firewall-cmd --reload
```

    # NOTE: Kibana on 443 tcp port *without* redirection needs additional permissions:

```
setcap 'CAP_NET_BIND_SERVICE=+eip' /usr/share/kibana/node/bin/node
```

- Cookie TTL and Cookie Keep Alive - for better work comfort, you can set two new variables in the Kibana configuration file `/etc/kibana/kibana.yml`:

```
login.cookiettl: 10
login.cookieKeepAlive: true
```

    CookieTTL is the value in minutes of the cookie's lifetime. The cookieKeepAlive renews this time with every valid query made by browser clicks.

    After saving changes in the configuration file, you must restart the service:

```
systemctl restart kibana
```

3.4 First login

If you log in to ITRS Log Analytics for the first time, you must specify the Index to be searched. We have the option of entering the name of your index, indicate a specific index from a given day, or using the asterix (*) to indicate all of them matching a specific index pattern. Therefore, to start working with ITRS Log Analytics application, we log in to it (by default the user: logserver/password:logserver).

After logging in to the application click the button "Set up index pattern" to add new index patter in Kibana:
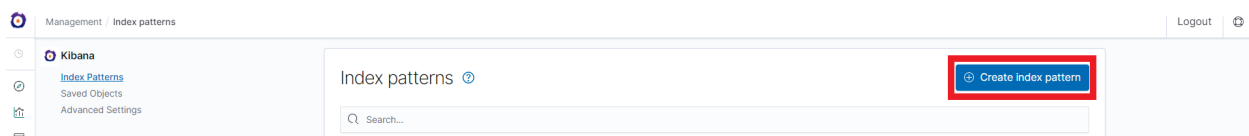
In the "Index pattern" field enter the name of the index or index pattern (after confirming that the index or sets of indexes exists) and click "Next step" button.
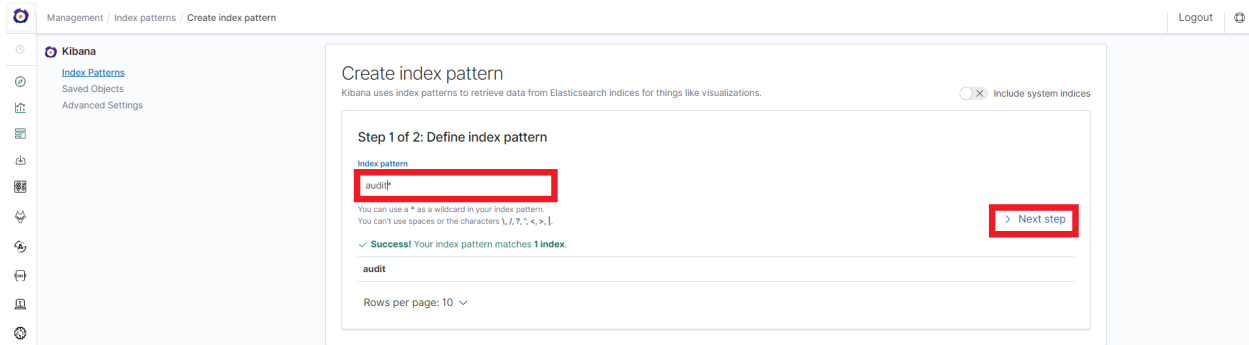
In the next step, from drop down menu select the "Time filter field name", after witch individual event (events) should be sorter. By default the *timestamp* is set, which is the time of occurrence of the event, but depending of the preferences. It may also be the time of the indexing or other selected based on the fields indicate on the event.
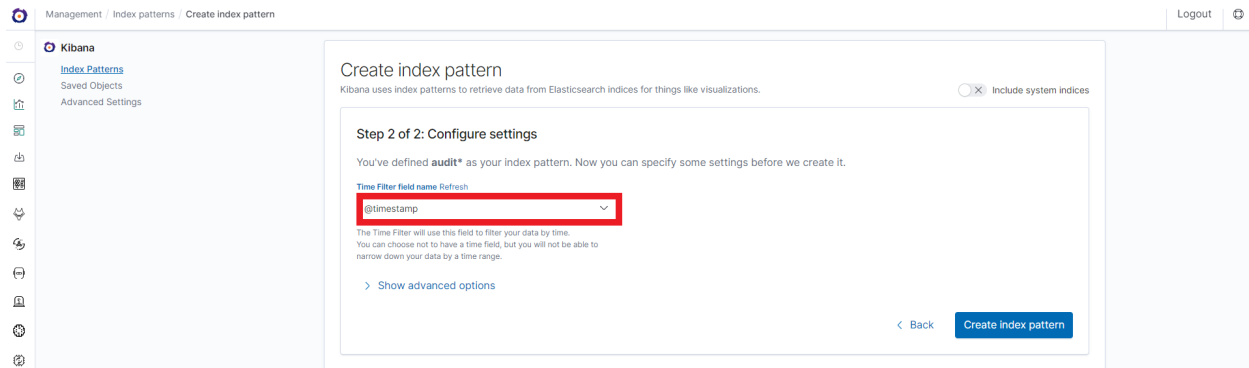
At any time, you can add more indexes or index patters by going to the main tab select „Management" and next select „Index Patterns".

3.5 Index selection

After login into ITRS Log Analytics you will going to „Discover" tab, where you can interactively explore your data. You have access to every document in every index that matches the selected index patterns.

If you want to change selected index, drop down menu with the name of the current object in the left panel. Clicking on the object from the expanded list of previously create index patterns, will change the searched index.
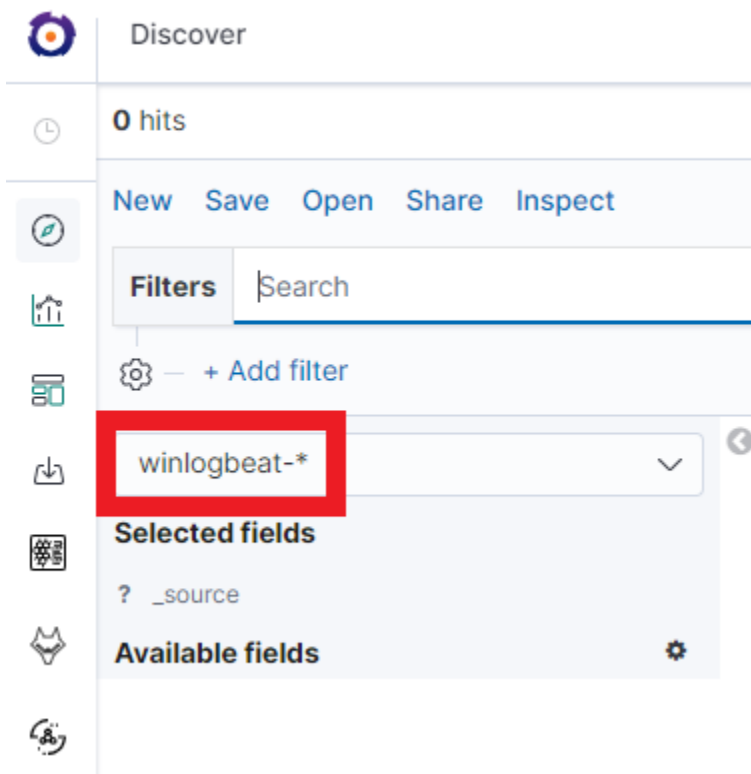
3.6 Changing default users for services

3.6.1 Change Kibana User

Edit file */etc/systemd/system/kibana.service*

```
        User=newuser
        Group= newuser
```

Edit */etc/default/kibana*

```
        user=" newuser "
        group=" newuser "
```

Add appropriate permission:

```
        chown newuser: /usr/share/kibana/ /etc/kibana/ -R
```

3.6.2 Change Elasticsearch User

Edit */usr/lib/tmpfiles.d/elasticsearch.conf* and change user name and group:

```
        d /var/run/elasticsearch 0755 newuser newuser -
```

Create directory:

```
        mkdir /etc/systemd/system/elasticsearch.service.d/
```

Edit */etc/systemd/system/elasticsearch.service.d/01-user.conf*

```
        [Service]
        User=newuser
        Group= newuser
```

Add appropriate permission:

```
        chown -R newuser: /var/lib/elasticsearch /usr/share/elasticsearch /etc/
→elasticsearch /var/log/elasticsearch
```

3.6.3 Change Logstash User

Create directory:

```
        mkdir /etc/systemd/system/logstash.service.d
```

Edit */etc/systemd/system/logstash.service.d/01-user.conf*

```
        [Service]
        User=newuser
        Group=newuser
```

Add appropriate permission:

```
        chown -R newuser: /etc/logstash /var/log/logstash
```

3.7 Custom installation the ITRS Log Analytics

If you need to install ITRS Log Analytics in non-default location, use the following steps.

1. Define the variable INSTALL_PATH if you do not want default paths like "/"

```
export INSTALL_PATH="/"
```

2. Install the firewalld service

```
yum install firewalld
```

3. Configure the firewalld service

```
systemctl enable firewalld
systemctl start firewalld
firewall-cmd --zone=public --add-port=22/tcp --permanent
firewall-cmd --zone=public --add-port=443/tcp --permanent
firewall-cmd --zone=public --add-port=5601/tcp --permanent
firewall-cmd --zone=public --add-port=9200/tcp --permanent
firewall-cmd --zone=public --add-port=9300/tcp --permanent
firewall-cmd --reload
```

4. Install and enable the epel repository

```
yum install epel-release
```

5. Install the Java OpenJDK

```
yum install java-1.8.0-openjdk-headless.x86_64
```

6. Install the reports dependencies, e.g. for mail and fonts

```
yum install fontconfig freetype freetype-devel fontconfig-devel libstdc++ urw-
→fonts net-tools ImageMagick ghostscript poppler-utils
```

7. Create the nessesery users acounts

```
useradd -M -d ${INSTALL_PATH}/usr/share/kibana -s /sbin/nologin kibana
useradd -M -d ${INSTALL_PATH}/usr/share/elasticsearch -s /sbin/nologin⎵
→elasticsearch
useradd -M -d ${INSTALL_PATH}/opt/alert -s /sbin/nologin alert
```

8. Remove .gitkeep files from source directory

```
find . -name ".gitkeep" -delete
```

9. Install the Elasticsearch 6.2.4 files

```
/bin/cp -rf elasticsearch/elasticsearch-6.2.4/* ${INSTALL_PATH}/
```

10. Install the Kibana 6.2.4 files

```
/bin/cp -rf kibana/kibana-6.2.4/* ${INSTALL_PATH}/
```

11. Configure the Elasticsearch system dependencies

```
/bin/cp -rf system/limits.d/30-elasticsearch.conf /etc/security/limits.d/
/bin/cp -rf system/sysctl.d/90-elasticsearch.conf /etc/sysctl.d/
/bin/cp -rf system/sysconfig/elasticsearch /etc/sysconfig/
/bin/cp -rf system/rsyslog.d/intelligence.conf /etc/rsyslog.d/
echo -e "RateLimitInterval=0\nRateLimitBurst=0" >> /etc/systemd/journald.conf
systemctl daemon-reload
systemctl restart rsyslog.service
sysctl -p /etc/sysctl.d/90-elasticsearch.conf
```

12. Configure the SSL Encryption for the Kibana

```
mkdir -p ${INSTALL_PATH}/etc/kibana/ssl
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -sha256 -subj '/CN=LOGSERVER/
↪subjectAltName=LOGSERVER/' -keyout ${INSTALL_PATH}/etc/kibana/ssl/kibana.key -
↪out ${INSTALL_PATH}/etc/kibana/ssl/kibana.crt
```

13. Install the Elasticsearch-auth plugin

```
cp -rf elasticsearch/elasticsearch-auth ${INSTALL_PATH}/usr/share/elasticsearch/
↪plugins/elasticsearch-auth
```

14. Install the Elasticsearh configuration files

```
/bin/cp -rf elasticsearch/*.yml ${INSTALL_PATH}/etc/elasticsearch/
```

15. Install the Elasticsicsearch system indices

```
mkdir -p ${INSTALL_PATH}/var/lib/elasticsearch
/bin/cp -rf elasticsearch/nodes ${INSTALL_PATH}/var/lib/elasticsearch/
```

16. Add necessary permission for the Elasticsearch directories

```
chown -R elasticsearch:elasticsearch ${INSTALL_PATH}/usr/share/elasticsearch $
↪{INSTALL_PATH}/etc/elasticsearch ${INSTALL_PATH}/var/lib/elasticsearch $
↪{INSTALL_PATH}/var/log/elasticsearch
```

17. Install the Kibana plugins

```
cp -rf kibana/plugins/* ${INSTALL_PATH}/usr/share/kibana/plugins/
```

18. Extrace the node_modules for plugins and remove archive

```
tar -xf ${INSTALL_PATH}/usr/share/kibana/plugins/node_modules.tar -C ${INSTALL_
↪PATH}/usr/share/kibana/plugins/
/bin/rm -rf ${INSTALL_PATH}/usr/share/kibana/plugins/node_modules.tar
```

19. Install the Kibana reports binaries

```
cp -rf kibana/export_plugin/* ${INSTALL_PATH}/usr/share/kibana/bin/
```

20. Create directory for the Kibana reports

```
/bin/cp -rf kibana/optimize ${INSTALL_PATH}/usr/share/kibana/
```

21. Install the python dependencies for reports

```
tar -xf kibana/python.tar -C /usr/lib/python2.7/site-packages/
```

22. Install the Kibana custom sources

```
/bin/cp -rf kibana/src/* ${INSTALL_PATH}/usr/share/kibana/src/
```

23. Install the Kibana configuration

```
/bin/cp -rf kibana/kibana.yml ${INSTALL_PATH}/etc/kibana/kibana.yml
```

24. Generate the iron secret salt for Kibana

```
echo "server.ironsecret: \"$(</dev/urandom tr -dc _A-Z-a-z-0-9 | head -c32)\"" >>
→ ${INSTALL_PATH}/etc/kibana/kibana.yml
```

25. Remove old cache files

```
rm -rf ${INSTALL_PATH}/usr/share/kibana/optimize/bundles/*
```

26. Install the Alert plugin

```
mkdir -p ${INSTALL_PATH}/opt
/bin/cp -rf alert ${INSTALL_PATH}/opt/alert
```

27. Install the AI plugin

```
/bin/cp -rf ai ${INSTALL_PATH}/opt/ai
```

28. Set the proper permissions

```
chown -R elasticsearch:elasticsearch ${INSTALL_PATH}/usr/share/elasticsearch/
chown -R alert:alert ${INSTALL_PATH}/opt/alert
chown -R kibana:kibana ${INSTALL_PATH}/usr/share/kibana ${INSTALL_PATH}/opt/ai $
→{INSTALL_PATH}/opt/alert/rules ${INSTALL_PATH}/var/lib/kibana
chmod -R 755 ${INSTALL_PATH}/opt/ai
chmod -R 755 ${INSTALL_PATH}/opt/alert
```

29. Install service files for the Alert, Kibana and the Elasticsearch

```
/bin/cp -rf system/alert.service /usr/lib/systemd/system/alert.service
/bin/cp -rf kibana/kibana-6.2.4/etc/systemd/system/kibana.service /usr/lib/
→systemd/system/kibana.service
/bin/cp -rf elasticsearch/elasticsearch-6.2.4/usr/lib/systemd/system/
→elasticsearch.service /usr/lib/systemd/system/elasticsearch.service
```

30. Set property paths in service files ${INSTALL_PATH}

```
perl -pi -e 's#/opt#'${INSTALL_PATH}'/opt#g' /usr/lib/systemd/system/alert.
→service
perl -pi -e 's#/etc#'${INSTALL_PATH}'/etc#g' /usr/lib/systemd/system/kibana.
→service
perl -pi -e 's#/usr#'${INSTALL_PATH}'/usr#g' /usr/lib/systemd/system/kibana.
→service
```

(continues on next page)

```
perl -pi -e 's#ES_HOME=#ES_HOME='${INSTALL_PATH}'#g' /usr/lib/systemd/system/
↪elasticsearch.service
perl -pi -e 's#ES_PATH_CONF=#ES_PATH_CONF='${INSTALL_PATH}'#g' /usr/lib/systemd/
↪system/elasticsearch.service
perl -pi -e 's#ExecStart=#ExecStart='${INSTALL_PATH}'#g' /usr/lib/systemd/system/
↪elasticsearch.service
```

31. Enable the system services

```
systemctl daemon-reload
systemctl reenable alert
systemctl reenable kibana
systemctl reenable elasticsearch
```

32. Set location for Elasticsearch data and logs files in configuration file

- Elasticsearch

```
perl -pi -e 's#path.data: #path.data: '${INSTALL_PATH}'#g' ${INSTALL_PATH}/
↪etc/elasticsearch/elasticsearch.yml
perl -pi -e 's#path.logs: #path.logs: '${INSTALL_PATH}'#g' ${INSTALL_PATH}/
↪etc/elasticsearch/elasticsearch.yml
perl -pi -e 's#/usr#'${INSTALL_PATH}'/usr#g' ${INSTALL_PATH}/etc/
↪elasticsearch/jvm.options
perl -pi -e 's#/usr#'${INSTALL_PATH}'/usr#g' /etc/sysconfig/elasticsearch
```

- Kibana

```
perl -pi -e 's#/etc#'${INSTALL_PATH}'/etc#g' ${INSTALL_PATH}/etc/kibana/
↪kibana.yml
perl -pi -e 's#/opt#'${INSTALL_PATH}'/opt#g' ${INSTALL_PATH}/etc/kibana/
↪kibana.yml
perl -pi -e 's#/usr#'${INSTALL_PATH}'/usr#g' ${INSTALL_PATH}/etc/kibana/
↪kibana.yml
```

- AI

```
perl -pi -e 's#/opt#'${INSTALL_PATH}'/opt#g' ${INSTALL_PATH}/opt/ai/bin/
↪conf.cfg
```

33. What next ?

- Upload License file to ${INSTALL_PATH}/usr/share/elasticsearch/directory.

- Setup cluster in ${INSTALL_PATH}/etc/elasticsearch/elasticsearch.yml

```
discovery.zen.ping.unicast.hosts: [ "172.10.0.1:9300", "172.10.0.2:9300" ]
```

- Redirect GUI to 443/tcp

```
        firewall-cmd --zone=public --add-masquerade --permanent
        firewall-cmd --zone=public --add-forward-
↪port=port=443:proto=tcp:toport=5601 --permanent
        firewall-cmd --reload
```

3.8 Plugins management in the Elasticsearch

Base installation of the ITRS Log Analytics contains the *elasticsearch-auth* plugin. You can extend the basic Elasticsearch functionality by installing the custom plugins.

Plugins contain JAR files, but may also contain scripts and config files, and must be installed on every node in the cluster.

After installation, each node must be restarted before the plugin becomes visible.

The Elasticsearch provides two categories of plugins:

- Core Plugins - it is plugins that are part of the Elasticsearch project.

- Community contributed - it is plugins that are external to the Elasticsearch project

3.8.1 Installing Plugins

Core Elasticsearch plugins can be installed as follows:

```
cd /usr/share/elasticsearch/
bin/elasticsearch-plugin install [plugin_name]
```

Example:

```
bin/elasticsearch-plugin install ingest-geoip

-> Downloading ingest-geoip from elastic
[=============================================] 100%
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@     WARNING: plugin requires additional permissions @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
* java.lang.RuntimePermission accessDeclaredMembers
* java.lang.reflect.ReflectPermission suppressAccessChecks
See http://docs.oracle.com/javase/8/docs/technotes/guides/security/permissions.html
for descriptions of what these permissions allow and the associated risks.

Continue with installation? [y/N]y
-> Installed ingest-geoip
```

Plugins from custom link or filesystem can be installed as follow:

```
cd /usr/share/elasticsearch/
sudo bin/elasticsearch-plugin install [url]
```

Example:

```
sudo bin/elasticsearch-plugin install file:///path/to/plugin.zip
bin\elasticsearch-plugin install file:///C:/path/to/plugin.zip
sudo bin/elasticsearch-plugin install http://some.domain/path/to/plugin.zip
```

3.8.2 Listing plugins

Listing currently loaded plugins

```
sudo bin/elasticsearch-plugin list
```

listing currently available core plugins:

```
sudo bin/elasticsearch-plugin list --help
```

3.8.3 Removing plugins

```
sudo bin/elasticsearch-plugin remove [pluginname]
```

3.8.4 Updating plugins

```
sudo bin/elasticsearch-plugin remove [pluginname]
sudo bin/elasticsearch-plugin install [pluginname]
```

3.9 ROOTless management

To configure ITRS Log Analytics so its services can be managed without root access follow these steps:

1. Create a file in /etc/sudoers.d (eg.: 10-logserver) with the content

```
%kibana ALL=/bin/systemctl status kibana
%kibana ALL=/bin/systemctl status kibana.service
%kibana ALL=/bin/systemctl stop kibana
%kibana ALL=/bin/systemctl stop kibana.service
%kibana ALL=/bin/systemctl start kibana
%kibana ALL=/bin/systemctl start kibana.service
%kibana ALL=/bin/systemctl restart kibana
%kibana ALL=/bin/systemctl restart kibana.service

%elasticsearch ALL=/bin/systemctl status elasticsearch
%elasticsearch ALL=/bin/systemctl status elasticsearch.service
%elasticsearch ALL=/bin/systemctl stop elasticsearch
%elasticsearch ALL=/bin/systemctl stop elasticsearch.service
%elasticsearch ALL=/bin/systemctl start elasticsearch
%elasticsearch ALL=/bin/systemctl start elasticsearch.service
%elasticsearch ALL=/bin/systemctl restart elasticsearch
%elasticsearch ALL=/bin/systemctl restart elasticsearch.service

%alert ALL=/bin/systemctl status alert
%alert ALL=/bin/systemctl status alert.service
%alert ALL=/bin/systemctl stop alert
%alert ALL=/bin/systemctl stop alert.service
%alert ALL=/bin/systemctl start alert
%alert ALL=/bin/systemctl start alert.service
%alert ALL=/bin/systemctl restart alert
%alert ALL=/bin/systemctl restart alert.service

%logstash ALL=/bin/systemctl status logstash
%logstash ALL=/bin/systemctl status logstash.service
%logstash ALL=/bin/systemctl stop logstash
```

(continues on next page)

```
%logstash ALL=/bin/systemctl stop logstash.service
%logstash ALL=/bin/systemctl start logstash
%logstash ALL=/bin/systemctl start logstash.service
%logstash ALL=/bin/systemctl restart logstash
%logstash ALL=/bin/systemctl restart logstash.service
```

2. Change permissions for files and directories

- Kibana, Elasticsearch, Alert

```
chmod g+rw /etc/kibana/kibana.yml /opt/alert/config.yaml /opt/ai/bin/conf.cfg /
→etc/elasticsearch/{elasticsearch.yml,jvm.options,log4j2.properties,properties.
→yml,role-mappings.yml}
        chmod g+rwx /etc/kibana/ssl /etc/elasticsearch/ /opt/{ai,alert} /opt/ai/
→bin
        chown -R elasticsearch:elasticsearch /etc/elasticsearch/
        chown -R kibana:kibana /etc/kibana/ssl
```

- Logstash

```
```bash
```

find /etc/logstash -type f -exec chmod g+rw {} ; find /etc/logstash -type d -exec chmod g+rwx {} ; chown -R logstash:logstash /etc/logstash

1. Add a user to groups defined earlier

```
usermod -a -G kibana,alert,elasticsearch,logstash service_user
```

From now on this user should be able to start/stop/restart services and modify configurations files.

3.10 ITRS Log Analytics Elasticsearch encryption

3.10.1 Generating Certificates

1. Requirements for certificate configuration:

   - **To encrypt traffic (HTTP and transport layer) of Elasticsearch you have to generate certificate authority which will be used to sign each node certificate of a cluster.**

   - **Elasticsearch certificate has to be generated in pkcs8 RSA format.**

2. Example certificate configuration (Certificates will be valid for 10 years based on this example):

```
# To make this process easier prepare some variables:
DOMAIN=loganalytics-node.test
DOMAIN_IP=10.4.3.185 # This is required if certificate validation is used on trasport
→layer
COUNTRYNAME=PL
STATE=Poland
COMPANY=LOGTEST
```

```
# Generate CA key:
openssl genrsa -out rootCA.key 4096

# Create and sign root certificate:
echo -e "${COUNTRYNAME}\n${STATE}\n\n${COMPANY}\n\n\n\n" | openssl req -x509 -new -
→nodes -key rootCA.key -sha256 -days 3650 -out rootCA.crt

# Crete RSA key for domain:
openssl genrsa -out ${DOMAIN}.pre 2048

# Convert generated key to pkcs8 RSA key for domain hostname
# (if you do not want to encrypt the key add "-nocrypt" at the end of the command;␣
→otherwise it will be necessary to add this password later in every config file):
openssl pkcs8 -topk8 -inform pem -in ${DOMAIN}.pre -outform pem -out ${DOMAIN}.key

# Create a Certificate Signing Request (openssl.cnf can be in a different location;␣
→this is the default for CentOS 7.7):
openssl req -new -sha256 -key ${DOMAIN}.key -subj "/C=PL/ST=Poland/O=EMCA/CN=${DOMAIN}
→" -reqexts SAN -config <(cat /etc/pki/tls/openssl.cnf <(printf
→"[SAN]\nsubjectAltName=DNS:${DOMAIN},IP:${DOMAIN_IP}")) -out ${DOMAIN}.csr

# Generate Domain Certificate
openssl x509 -req -in ${DOMAIN}.csr -CA rootCA.crt -CAkey rootCA.key -CAcreateserial -
→out ${DOMAIN}.crt -sha256 -extfile <(printf "[req]\ndefault_
→bits=2048\ndistinguished_name=req_distinguished_name\nreq_extensions=req_ext\n[req_
→distinguished_name]\ncountryName=${COUNTRYNAME}\nstateOrProvinceName=${STATE}
→\norganizationName=${COMPANY}\ncommonName=${DOMAIN}\n[req_ext]\nsubjectAltName=@alt_
→names\n[alt_names]\nDNS.1=${DOMAIN}\nIP=${DOMAIN_IP}\n") -days 3650 -extensions req_
→ext

# Verify the validity of the generated certificate
openssl x509 -in ${DOMAIN}.crt -text -noout
```

1. Right now you should have these files:

```
$ ls -1 | sort
loganalytics-node.test.crt
loganalytics-node.test.csr
loganalytics-node.test.key
loganalytics-node.test.pre
rootCA.crt
rootCA.key
rootCA.srl
```

1. Create a directory to store required files (users: elasticsearch, kibana and logstash have to be able to read these files):

```
mkdir /etc/elasticsearch/ssl
cp {loganalytics-node.test.crt,loganalytics-node.test.key,rootCA.crt} /etc/
→elasticsearch/ssl
chown -R elasticsearch:elasticsearch /etc/elasticsearch/ssl
chmod 755 /etc/elasticsearch/ssl
chmod 644 /etc/elasticsearch/ssl/*
```

3.10.2 Setting up configuration files

1. Append or uncomment below lines in `/etc/elasticsearch/elasticsearch.yml` and change paths to proper values (based on past steps):

```
## Transport layer encryption
logserverguard.ssl.transport.enabled: true
logserverguard.ssl.transport.pemcert_filepath: "/etc/elasticsearch/ssl/loganalytics-
→node.test.crt"
logserverguard.ssl.transport.pemkey_filepath: "/etc/elasticsearch/ssl/loganalytics-
→node.test.key"
logserverguard.ssl.transport.pemkey_password: "password_for_pemkey" # if there is no␣
→password leve ""
logserverguard.ssl.transport.pemtrustedcas_filepath: "/etc/elasticsearch/ssl/rootCA.
→crt"

logserverguard.ssl.transport.enforce_hostname_verification: true
logserverguard.ssl.transport.resolve_hostname: true

logserverguard.ssl.transport.enabled_ciphers:
 - "TLS_DHE_RSA_WITH_AES_256_GCM_SHA384"
logserverguard.ssl.transport.enabled_protocols:
 - "TLSv1.2"

## HTTP layer encryption
logserverguard.ssl.http.enabled: true
logserverguard.ssl.http.pemcert_filepath: "/etc/elasticsearch/ssl/loganalytics-node.
→test.crt"
logserverguard.ssl.http.pemkey_filepath: "/etc/elasticsearch/ssl/loganalytics-node.
→test.key"
logserverguard.ssl.http.pemkey_password: "password_for_pemkey" # if there is no␣
→password leve ""
logserverguard.ssl.http.pemtrustedcas_filepath: "/etc/elasticsearch/ssl/rootCA.crt"

logserverguard.ssl.http.clientauth_mode: OPTIONAL
logserverguard.ssl.http.enabled_ciphers:
 - "TLS_DHE_RSA_WITH_AES_256_GCM_SHA384"

logserverguard.ssl.http.enabled_protocols:
 - "TLSv1.2"
```

1. Append or uncomment below lines in `/etc/kibana/kibana.yml` and change paths to proper values:

```
# For below two, both IP or HOSTNAME (https://loganalytics-node.test:PORT) can be␣
→used because IP has been supplied in "alt_names"
elasticsearch.url: "https://10.4.3.185:8000" # Default is "http://localhost:8000"
---
elastfilter.url: "https://10.4.3.185:9200" # Default is"http://localhost:9200"
---
# Elasticsearch trafic encryption
# There is also an option to use "127.0.0.1/localhost" and to not supply path to CA.␣
→Verification Mode should be then changed to "none".
elasticsearch.ssl.verificationMode: full
elasticsearch.ssl.certificate: "/etc/elasticsearch/ssl/loganalytics-node.test.crt"
elasticsearch.ssl.key: "/etc/elasticsearch/ssl/loganalytics-node.test.key"
elasticsearch.ssl.keyPassphrase: "password_for_pemkey" # this line is not required if␣
→there is no password
elasticsearch.ssl.certificateAuthorities: "/etc/elasticsearch/ssl/rootCA.crt"
```

1. Append or uncomment below lines in `/opt/alert/config.yaml` and change paths to proper values:

```
# Connect with TLS to Elasticsearch
use_ssl: True

# Verify TLS certificates
verify_certs: True

# Client certificate:
client_cert: /etc/elasticsearch/ssl/loganalytics-node.test.crt
client_key: /etc/elasticsearch/ssl/loganalytics-node.test.key
ca_certs: /etc/elasticsearch/ssl/rootCA.crt
```

1. For CSV/HTML export to work properly rootCA.crt generated in first step has to be "installed" on the server. Below example for CentOS 7:

```
# Copy rootCA.crt and update CA trust store
cp /etc/elasticsearch/ssl/rootCA.crt /etc/pki/ca-trust/source/anchors/rootCA.crt
update-ca-trust
```

1. Intelligence module. Generate pkcs12 keystore/cert:

```
DOMAIN=loganalytics-node.test
keytool -import -file /etc/elasticsearch/ssl/rootCA.crt -alias root -keystore root.jks
openssl pkcs12 -export -in /etc/elasticsearch/ssl/${DOMAIN}.crt -inkey /etc/
→elasticsearch/ssl/${DOMAIN}.key -out ${DOMAIN}.p12 -name "${DOMAIN}" -certfile /etc/
→elasticsearch/ssl/rootCA.crt
```

```
# Configure /opt/ai/bin/conf.cfg
https_keystore=/path/to/pk12/loganalytics-node.test.p12
https_truststore=/path/to/root.jks
https_keystore_pass=bla123
https_truststore_pass=bla123
```

3.10.3 Logstash/Beats

You can eather install CA to allow Logstash and Beats traffic or you can supply required certificates in config:

1. Logstash:

```
output {
  elasticsearch {
    hosts => "https://loganalytics-node.test:9200"
    ssl => true
    index => "winlogbeat-%{+YYYY.MM}"
    user => "logstash"
    password => "logstash"
    cacert => "/path/to/cacert/rootCA.crt"
  }
}
```

1. Beats:

```
output.elasticsearch.hosts: ["https://loganalytics-node.test:9200"]
output.elasticsearch.protocol: "https"
output.elasticsearch.ssl.enabled: true
output.elasticsearch.ssl.certificate_authorities: ["/path/to/cacert/rootCA.crt"]
```

Additionally, for any beats program to be able to write to elasticsearch, you will have to make changes to "enabled_ciphers" directive in "/etc/elasticsearch/elasticsearch.yml". This is done by commenting:

```
logserverguard.ssl.http.enabled_ciphers:
 - "TLS_DHE_RSA_WITH_AES_256_GCM_SHA384"
```

Otherwise, the beat will not be able to send documents directly and if you want to avoid it you can send a document with Logstash first.

Discovery

4.1 Time settings and refresh

In the upper right corner there is a section in which it defines the range of time that ITRS Log Analytics will search in terms of conditions contained in the search bar. The default value is the last 15 minutes.
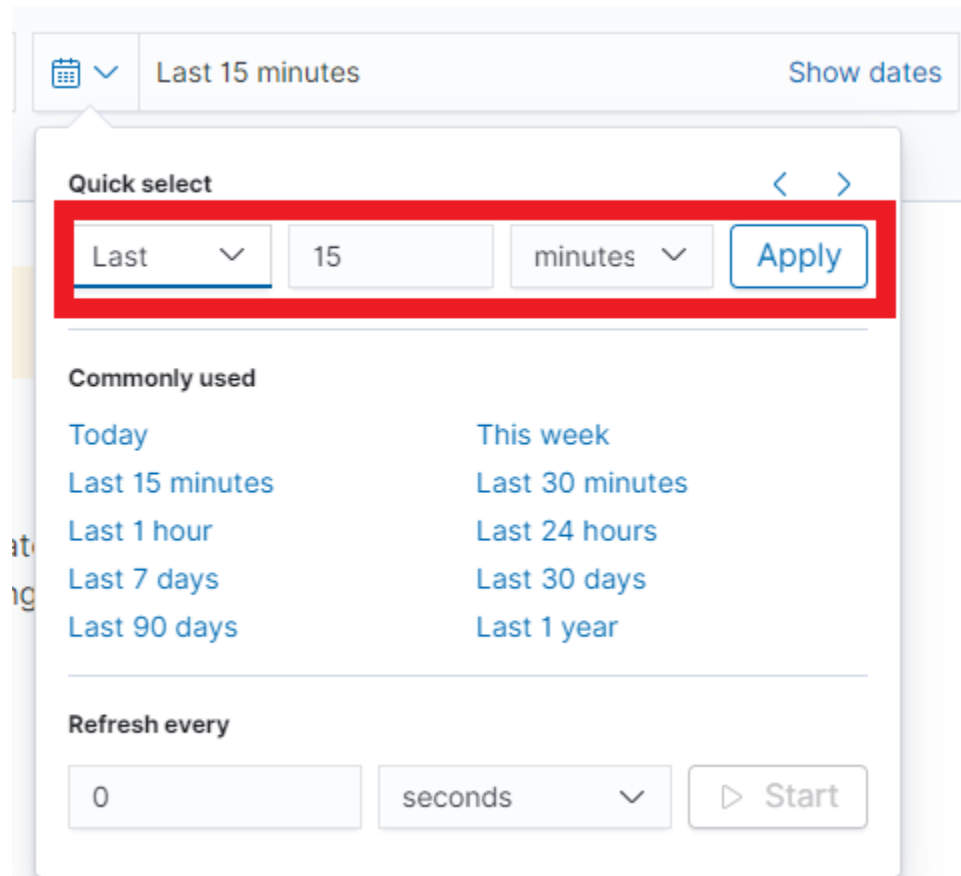
After clicking this selection, we can adjust the scope of search by selecting one of the three tabs in the drop-down window:

- **Quick**: contain several predefined ranges that should be clicked.
- **Relative**: in this windows specify the day from which ITRS Log Analytics should search for data.
- **Absolute**: using two calendars we define the time range for which the search results are to be returned.
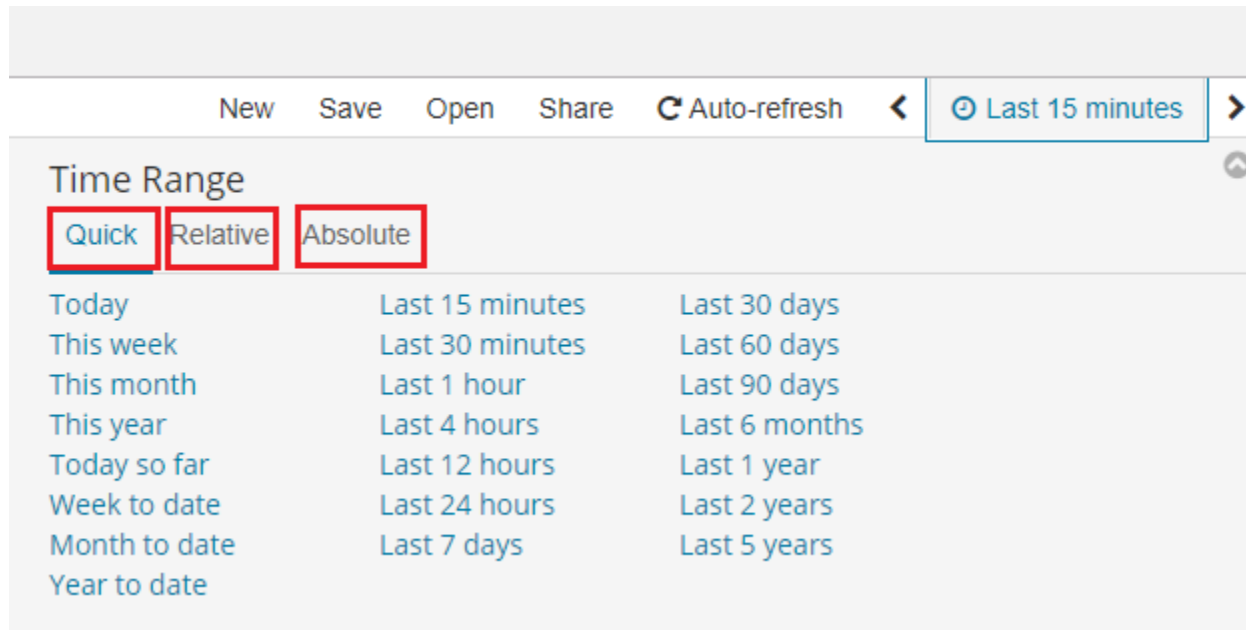
4.2 Fields

ITRS Log Analytics in the body of searched events, it recognize fields that can be used to created more precision queries. The extracted fields are visible in the left panel. They are divided on three types: timestamp, marked on clock icon ; text, marked with the letter "t" and digital, marked witch hashtag .

Pointing to them and clicking on icon , they are automatically transferred to the „Selected Fields" column and in the place of events a table with selected columns is created on regular basis. In the "Selected Fields" selection you can also delete specific fields from the table by clicking on the selected element.

Selected Fields

○ timestamp

ᴛ operation

ᴛ username

ᴛ _id

Available Fields ⚙

ᴛ _index

\# _score

ᴛ _type

ᴛ method

ᴛ params._

ᴛ params.allow_no_indices

ᴛ params.ignore_unavailable

ᴛ params.include_defaults

ᴛ params.level

ᴛ params.preference

ᴛ params.timeout

ᴛ request
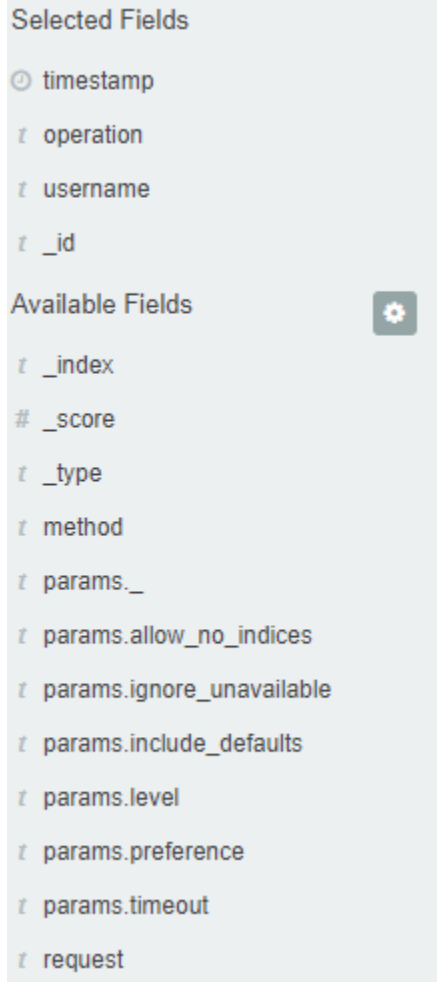
4.3 Filtering and syntax building

We use the query bar to search interesting events. For example, after entering the word „error", all events that contain the word will be displayed, additional highlighting them with an yellow background.
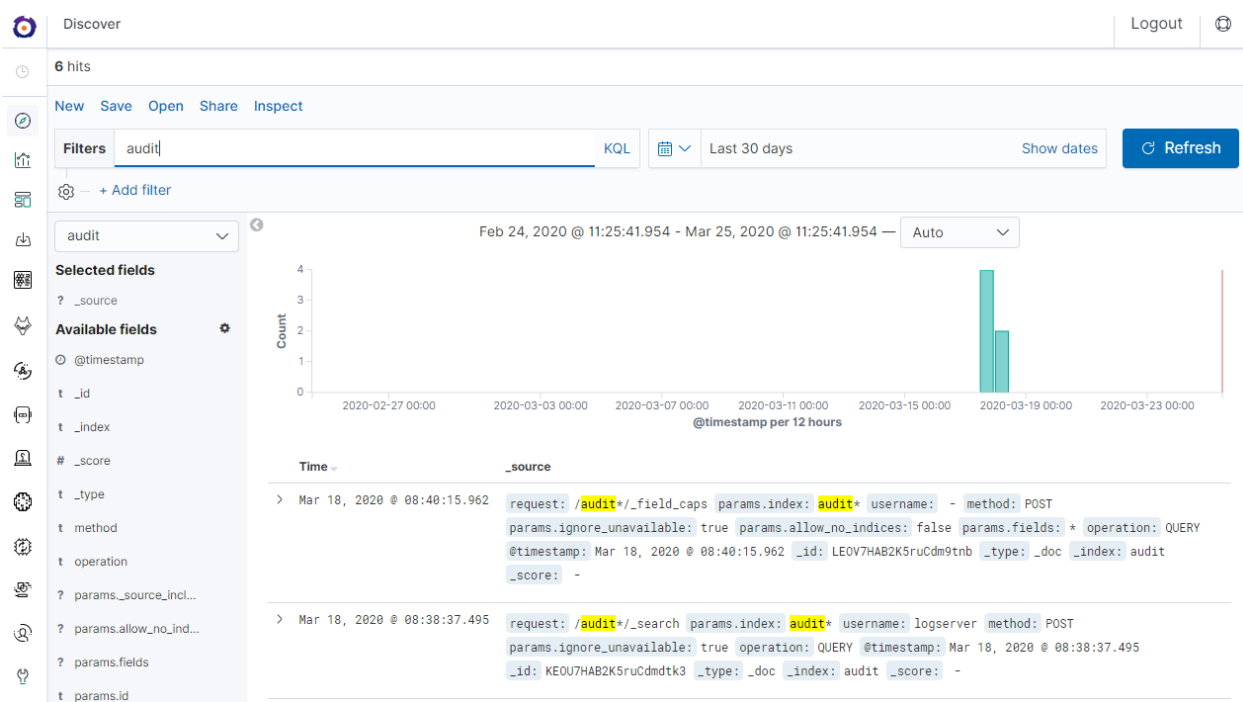
4.3.1 Syntax

Fields can be used in the similar way by defining conditions that interesting us. The syntax of such queries is:

```
<fields_name:<fields_value>
```

Example:

```
status:500
```

This query will display all events that contain the „status" fields with a value of 500.

4.3.2 Filters

The field value does not have to be a single, specific value. For digital fields we can specify range in the following scheme:

```
<fields_name:[<range_from TO <range_to]
```

Example:

```
status:[500 TO 599]
```

This query will return events with status fields that are in the range 500 to 599.

4.3.3 Operators

The search language used in ITRS Log Analytics allows to you use logical operators „AND", „OR" and „NOT", which are key and necessary to build more complex queries.

- **AND** is used to combined expressions, e.g. „error AND „access denied". If an event contain only one expression or the words error and denied but not the word access, then it will not be displayed.

- **OR** is used to search for the events that contain one OR other expression, e.g. „status:500" OR "denied". This query will display events that contain word „denied" or status field value of 500. ITRS Log Analytics uses this operator by default, so query „status:500" "denied" would return the same results.

- **NOT** is used to exclude the following expression e.g. „status:[500 TO 599] NOT status:505" will display all events that have a status fields, and the value of the field is between 500 and 599 but will eliminate from the result events whose status field value is exactly 505.

- **The above methods** can be combined with each other by building even more complex queries. Understanding how they work and joining it, is the basis for effective searching and full use of ITRS Log Analytics.

Example of query built from connected logical operations:

```
status:[500 TO 599] AND („access denied" OR error) NOT status:505
```

Returns in the results all events for which the value of status fields are in the range of 500 to 599, simultaneously contain the word „access denied" or „error", omitting those events for which the status field value is 505.

4.4 Saving and deleting queries

Saving queries enables you to reload and use them in the future.

4.4.1 Save query

To save query, click on the "Save" button under on the query bar:

New   Save   Open   Share

This will bring up a window in which we give the query a name and then click the button   Save   .

Save search   ×

Title

New Saved Search

Cancel   **Confirm Save**

Saved queries can be opened by going to „Open" from the main menu at the top of the page, and select saved search from the search list:

Additional you can use "Saved Searchers Filter.." to filter the search list.

4.4.2 Open query

To open a saved query from the search list, you can click on the name of the query you are interested in.

After clicking on the icon on the name of the saved query and chose "Edit Query DSL", we will gain access to the advanced editing mode, so that we can change the query on at a lower level.

It is a powerful tool designed for advanced users, designed to modify the query and the way it is presented by ITRS Log Analytics.

4.4.3 Delete query

To delete a saved query, open it from the search list, and then click on the button .

If you want delete many saved queries simultaneously go to the "Management Object" -> "Saved Object" -> "Searches" select it in the list (the icon to the left of the query name), and then click "Delete" button.

From this level, you can also export saved queries in the same way. To do this, you need to click on
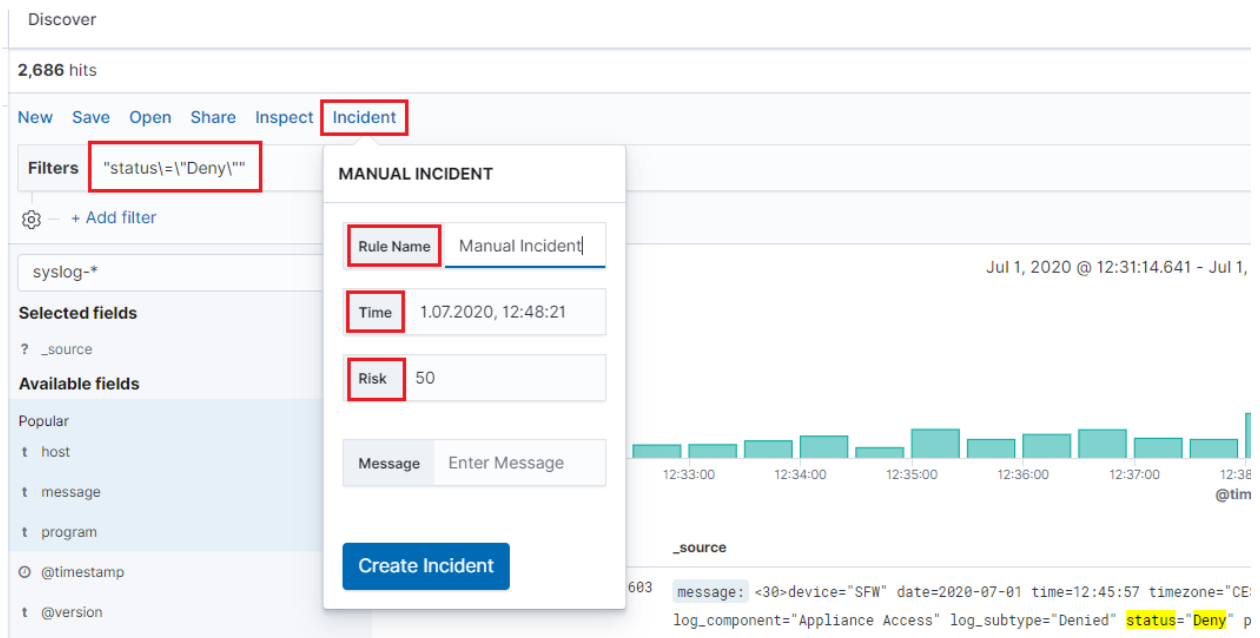
and choose the save location. The file will be saved in .JSON format. If you then want to import such a file to ITRS

Log Analytics, click on button ⊥ Import , at the top of the page and select the desired file.
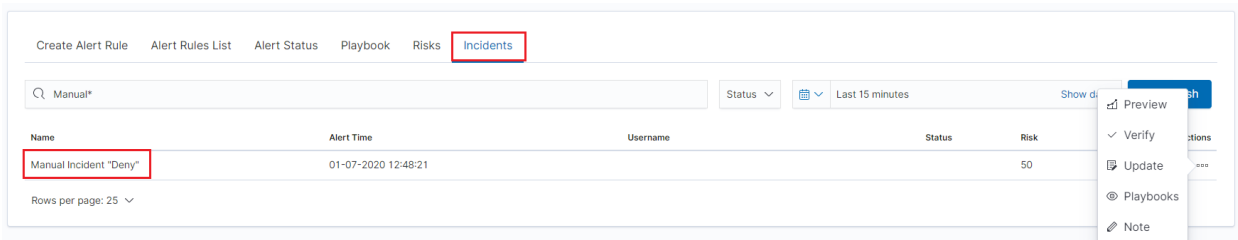
4.5 Manual incident

The `Discovery` module allows you to manually create incidents that are saved in the `Incidents` tab of the `Alerts` module. Manual incidents are based on search results or filtering. For a manual incident, you can save the following parameters:

- Rule name
- Time
- Risk
- Message

After saving the manual incident, you can go to the Incident tab in the Alert module to perform the incident handling procedure.

Visualizations

Visualize enables you to create visualizations of the data in your ITRS Log Analytics indices. You can then build dashboards that display related visualizations. Visualizations are based on ITRS Log Analytics queries. By using a series of ITRS Log Analytics aggregations to extract and process your data, you can create charts that show you the trends, spikes, and dips.

5.1 Creating visualization

5.1.1 Create

To create visualization, go to the „Visualize" tab from the main menu. A new page will be appearing where you can create or load visualization.

5.1.2 Load

To load previously created and saved visualization, you must select it from the list.

In order to create a new visualization, you should choose the preferred method of data presentation.

New Visualization

Q Filter

Select a visualization type

Start creating your visualization by
selecting a type for that visualization.

Area

Controls ᴱ

Coordinate Map

Data Table

Gauge

Goal

Heat Map

Horizontal Bar

Line

Markdown

Metric

Network

Pie

Region Map

TSVB

Tag Cloud

Timelion

Vega ᴱ

Vertical Bar

Next, specify whether the created visualization will be based on a new or previously saved query. If on new one, select the index whose visualization should concern. If visualization is created from a saved query, you just need to select the appropriate query from the list, or (if there are many saved searches) search for them by name.

5.2 Vizualization types

Before the data visualization will be created, first you have to choose the presentation method from an existing list. Currently there are five groups of visualization types. Each of them serves different purposes. If you want to see only the current number of products sold, it is best to choose „Metric", which presents one value.

However, if we would like to see user activity trends on pages in different hour and days, a better choice will be „Area chart", which displays a chart with time division.
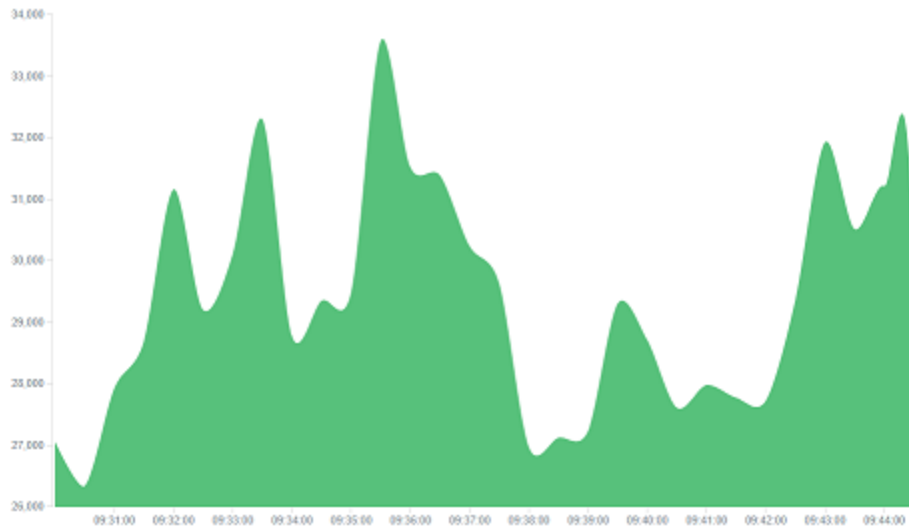
The „Markdown widget" views is used to place text e.g. information about the dashboard, explanations and instruction on how to navigate. Markdown language was used to format the text (the most popular use is GitHub). More information and instruction can be found at this link: https://help.github.com/categories/writing-on-github/

5.3 Edit visualization and saving

5.3.1 Edititing

Editing a saved visualization enables you to directly modify the object definition. You can change the object title, add a description, and modify the JSON that defines the object properties. After selecting the index and the method of data presentation, you can enter the editing mode. This will open a new window with empty visualization.
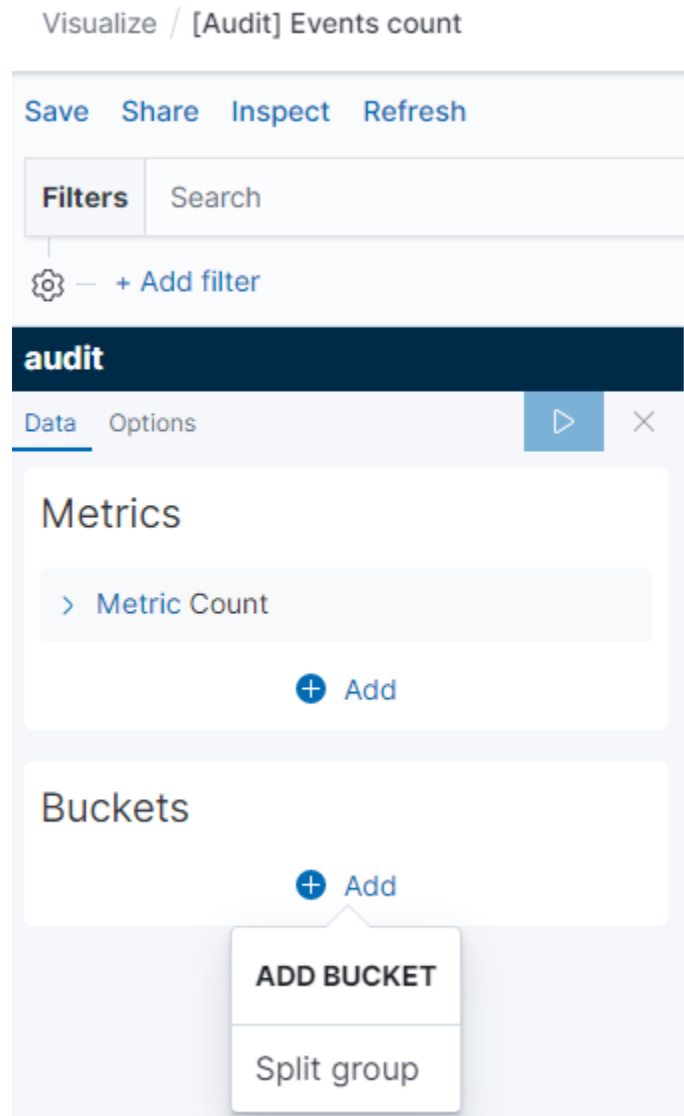
At the very top there is a bar of queries that cat be edited throughout the creation of the visualization. It work in the same way as in the "Discover" tab, which means searching the raw data, but instead of the data being displayed, the visualization will be edited. The following example will be based on the „Area chart". The visualization modification panel on the left is divided into three tabs: „Data", "Metric & Axes" and „Panel Settings".

In the „Data" tab, you can modify the elements responsible for which data and how should be presented. In this tab there are two sectors: "metrics", in which we set what data should be displayed, and „buckets" in which we specify how they should be presented.

Select the Metrics & Axes tab to change the way each individual metric is shown on the chart. The data series are styled in the Metrics section, while the axes are styled in the X and Y axis sections.

In the „Panel Settings" tab, there are settings relating mainly to visual aesthetics. Each type of visualization has separate options.

To create the first graph in the char modification panel, in the „Data" tab we add X-Axis in the "buckets" sections. In „Aggregation" choose „Histogram", in „Field" should automatically be located "timestamp" and "interval": "Auto" (if not, this is how we set it). Click on the icon on the panel. Now our first graph should show up.

Some of the options for „Area Chart" are:

Smooth Lines - is used to smooth the graph line.
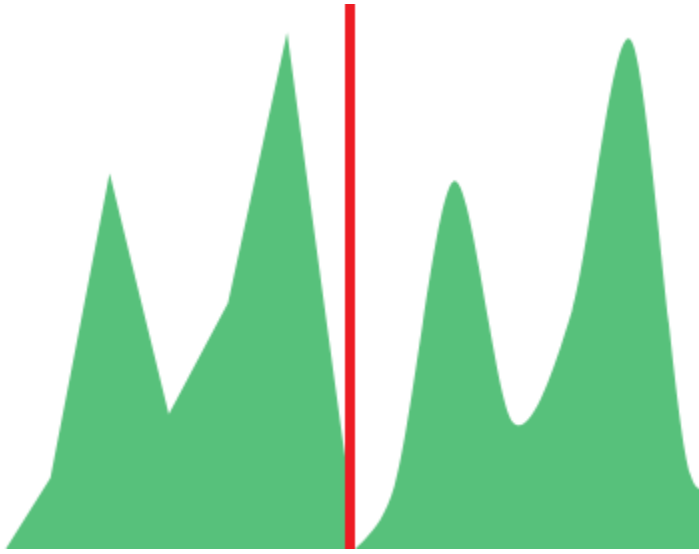
- **Current time marker** – places a vertical line on the graph that determines the current time.

- **Set Y-Axis Extents** – allows you to set minimum and maximum values for the Y axis, which increases the readability of the graphs. This is useful, if we know that the data will never be less then (the minimum value), or to indicate the goals the company (maximum value).

- **Show Tooltip** – option for displaying the information window under the mouse cursor, after pointing to the point on the graph.

5.3.2 Saving

To save the visualization, click on the "Save" button under on the query bar:

give it a name and click the button Save .

5.3.3 Load

To load the visualization, go to the "Management Object" -> "Saved Object" -> "Visualizations" select it from the list. From this place, we can also go into advanced editing mode. To view of the visualization use button.# Dashboards #

5.4 Dashboards

Dashboard is a collection of several visualizations or searches. Depending on how it is build and what visualization it contains, it can be designed for different teams e.g.:

- SOC - which is responsible for detecting failures or threats in the company;

- business - which thanks to the listings can determine the popularity of products and define the strategy of future sales and promotions;

- managers and directors - who may immediately have access to information about the performance units or branches.

5.4.1 Create

To create a dashboard from previously saved visualization and queries, go to the „Dashboard" tab in the main menu. When you open it, a new page will appear.
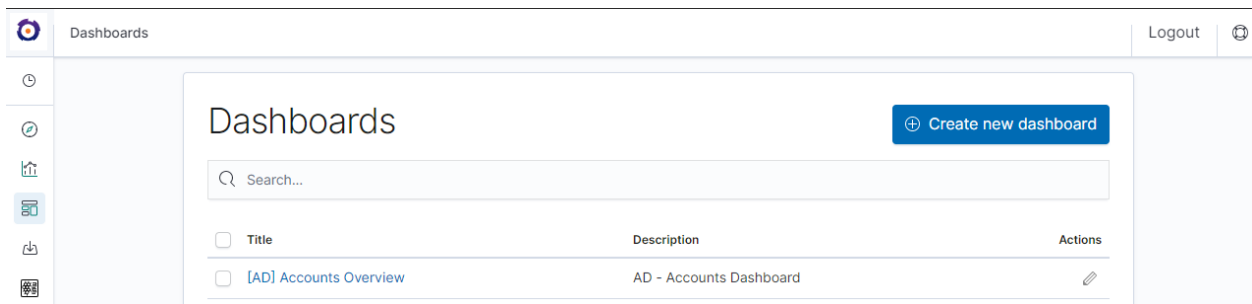
Clicking on the icon "Add" at the top of page select "Visualization" or "Saved Search" tab.
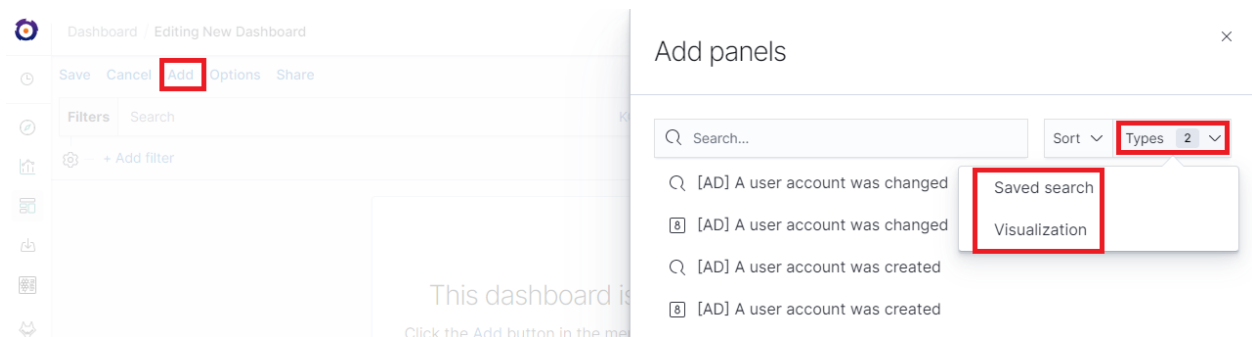
and selecting a saved query and / or visualization from the list will add them to the dashboard. If, there are a large number of saved objects, use the bar to search for them by name.
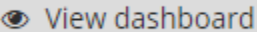
Elements of the dashboard can be enlarged arbitrarily (by clicking on the right bottom corner of object and dragging the border) and moving (by clicking on the title bar of the object and moving it).

5.4.2 Saving

To save a dashboard, click on the "Save" button to the up of the query bar and give it a name.

5.4.3 Load

To load the Dashboard, go to the "Management Object" -> "Saved Object" -> "Dashborad" select it from the list. From this place, we can also go into advanced editing mode. To view of the visualization use button.# Sharing dashboards #

5.5 Sharing dashboards

The dashboard can be share with other ITRS Log Analytics users as well as on any page - by placing a snippet of code. Provided that it cans retrieve information from ITRS Log Analytics.

To do this, create new dashboard or open the saved dashboard and click on the "Share" to the top of the page. A window will appear with generated two URL. The content of the first one "Embaded iframe" is used to provide the dashboard in the page code, and the second "Link" is a link that can be passed on to another user. There are two option for each, the first is to shorten the length of the link, and second on copies to clipboard the contest of the given bar.
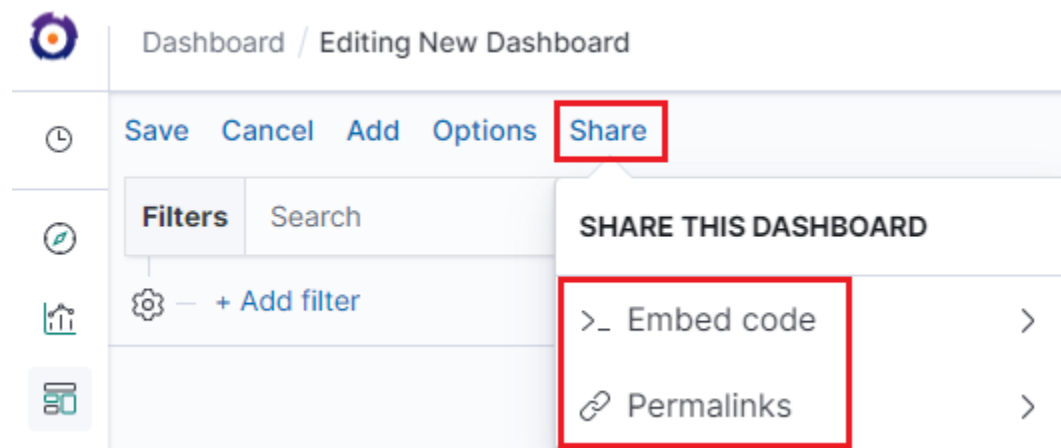
5.6 Dashboard drilldown

In discovery tab search for message of Your interest
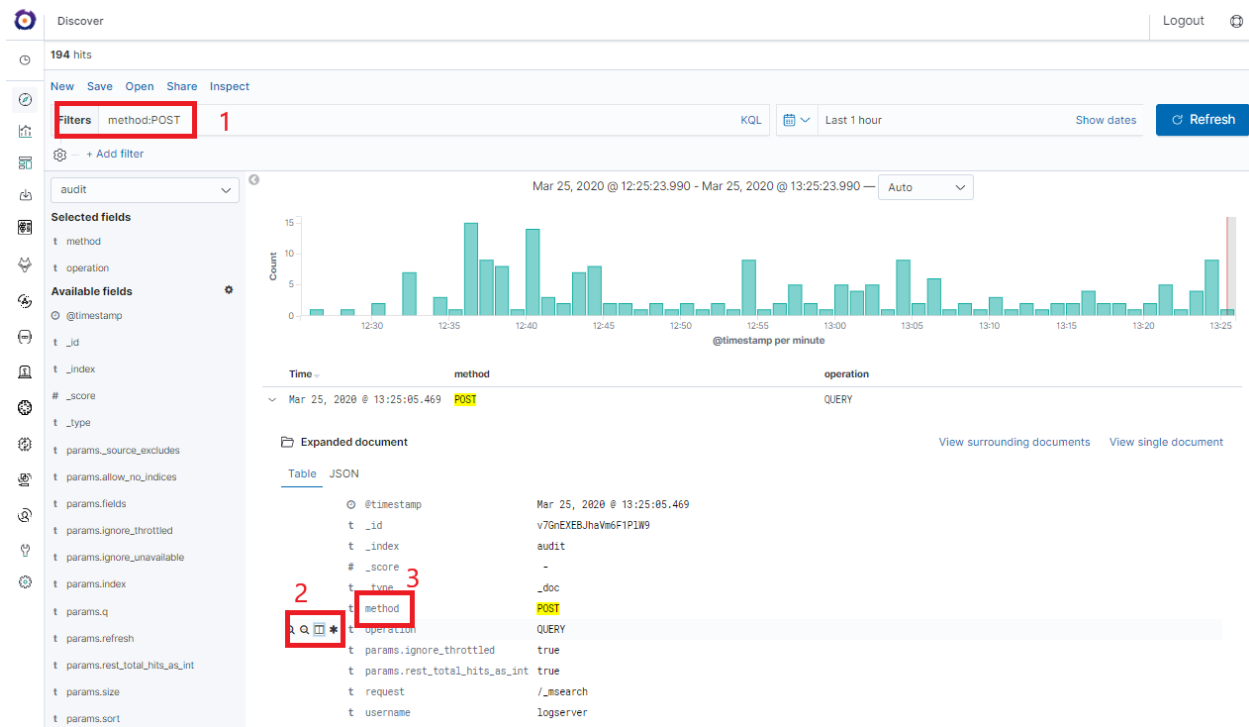
Save Your search
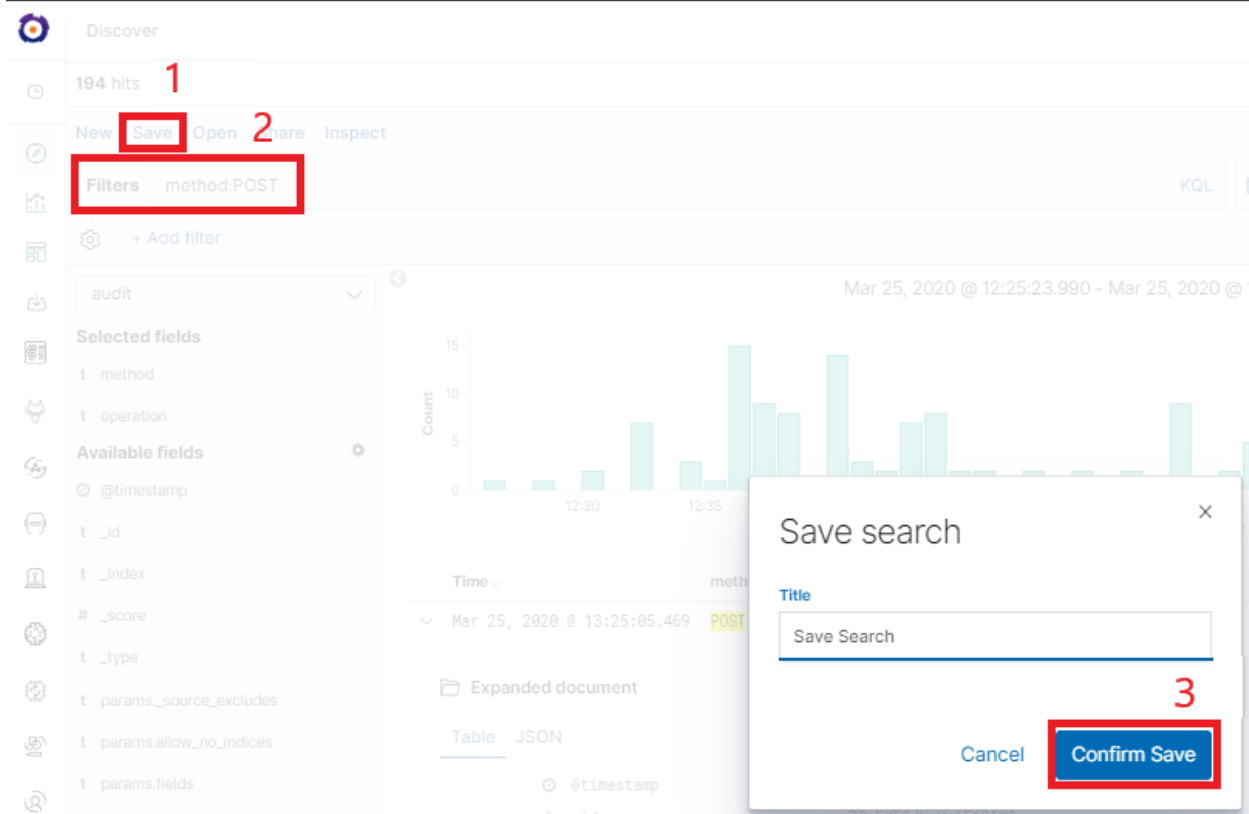
Check You „Shared link" and copy it
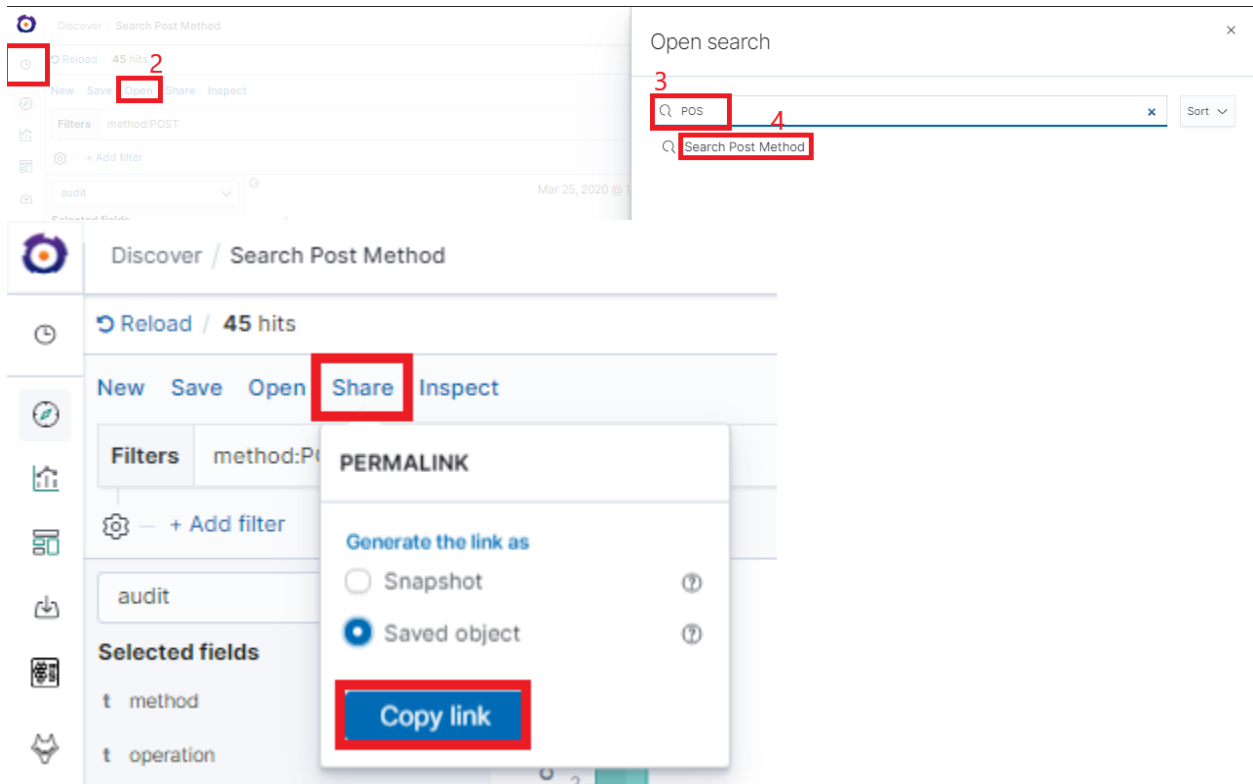
! ATTENTION ! Do not copy „?_g=()" at the end.

Select Alerting module
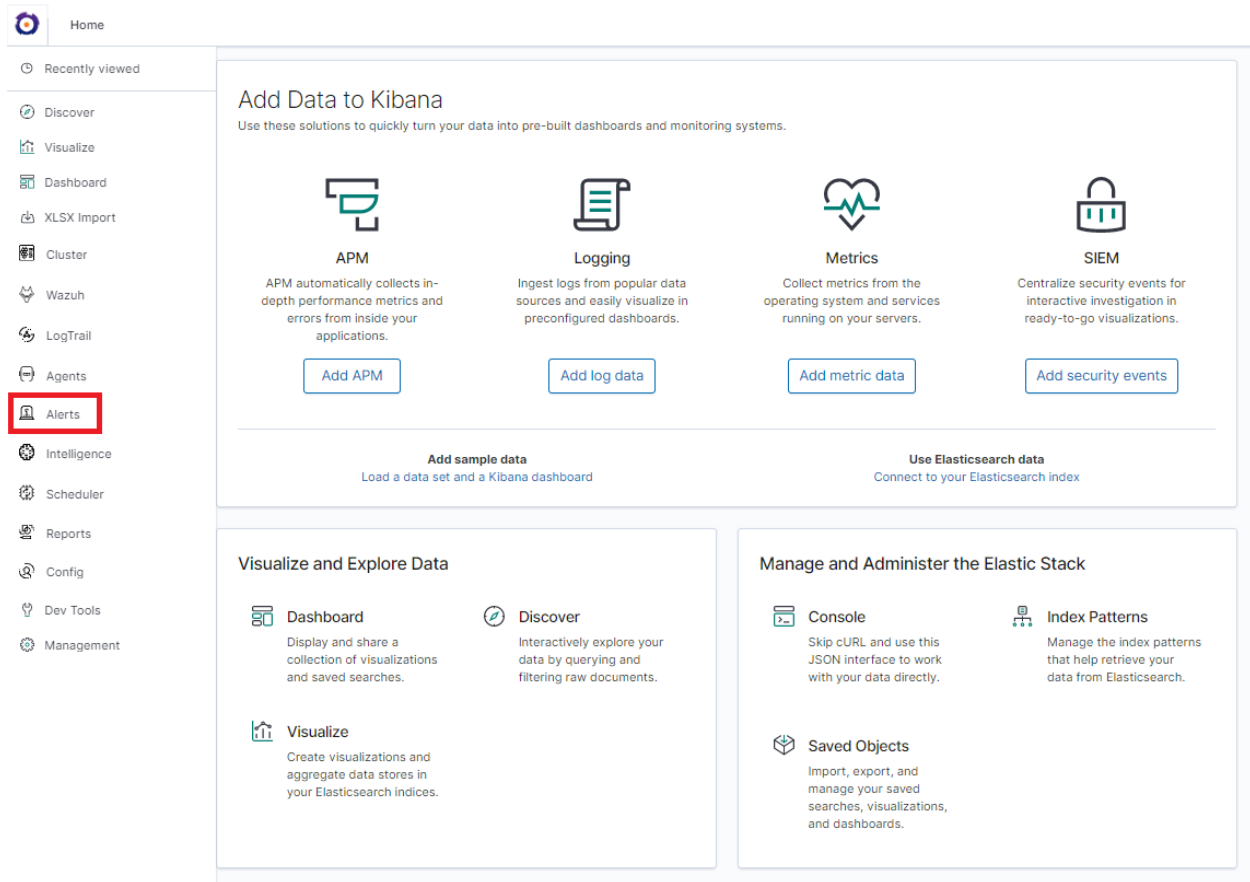
Once Alert is created use `ANY` frame to add the following directives:

```
Use_kibana4_dashboard: paste Your „shared link" here
```

`use_kibana_dashboard:` - The name of a Kibana dashboard to link to. Instead of generating a dashboard from a template, Alert can use an existing dashboard. It will set the time range on the dashboard to around the match time, upload it as a temporary dashboard, add a filter to the query_key of the alert if applicable, and put the url to the dashboard in the alert. (Optional, string, no default).

```
Kibana4_start_timedelta
```

`kibana4_start_timedelta:` Defaults to 10 minutes. This option allows you to specify the start time for the generated kibana4 dashboard. This value is added in front of the event. For example,

```
`kibana4_start_timedelta: minutes: 2`
```

```
Kibana4_end_timedelta`
```

`kibana4_end_timedelta:` Defaults to 10 minutes. This option allows you to specify the end time for the generated kibana4 dashboard. This value is added in back of the event. For example,

```
kibana4_end_timedelta: minutes: 2
```


Sample:

Search for triggered alert in Discovery tab. Use alert* search pattern.

Refresh the alert that should contain url for the dashboard. Once available, kibana_dashboard field can be exposed to dashboards giving You a real drill down feature.

Reports

ITRS Log Analytics contains a module for creating reports that can be run cyclically and contain only interesting data, e.g. a weekly sales report.
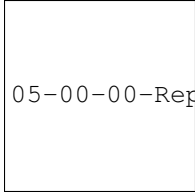
To go to the reports windows, select to tiles icon from the main menu bar, and then go to the „Reports" icon (To go back, go to the „Search" icon).
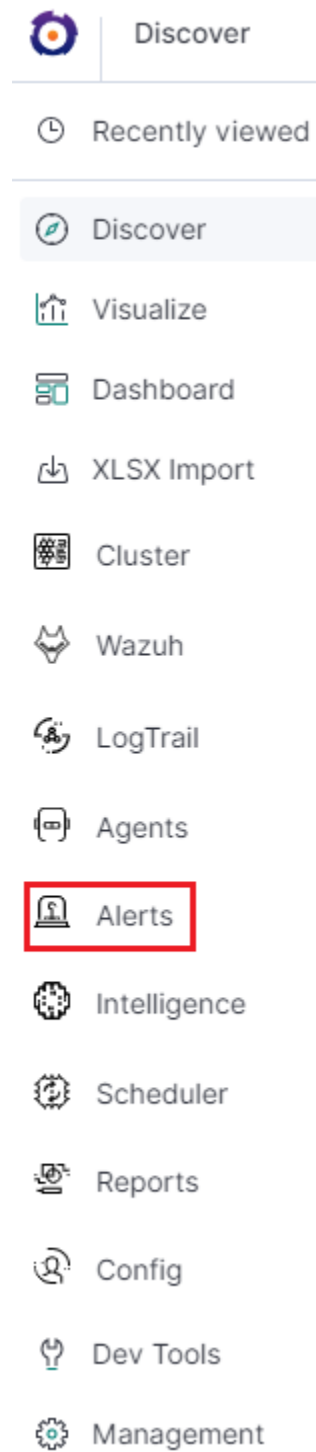
6.1 CSV Report

To export data to CSV Report click the Reports icon, you immediately go to the first tab - Export Task Management.

05-00-00-Reports/./media/media/image39_js.png

In this tab we have the opportunity to specify the source from which we want to do export. It can be an index pattern. After selecting it, we confirm the selection with the Submit button and a report is created at the moment. The symbol

Refresh List ⟳ can refresh the list of reports and see what its status is.

We can also create a report by pointing to a specific index from the drop-down list of indexes.

We can also check which fields are to be included in the report. The selection is confirmed by the Submit button.

When the process of generating the report (Status:Completed) is finished, we can download it (Download button) or delete (Delete button). The downloaded report in the form of *.csv file can be opened in the browser or saved to the disk.

In this tab, the downloaded data has a format that we can import into other systems for further analysis.

6.2 PDF Report

In the Export Dashboard tab we have the possibility to create graphic reports in PDF files. To create such a report, just from the drop-down list of previously created and saved Dashboards, indicate the one we are interested in, and then confirm the selection with the Submit button. A newly created export with the Processing status will appear on the list under Dashboard Name. When the processing is completed, the Status changes to Complete and it will be possible to download the report.

By clicking the Download button, the report is downloaded to the disk or we can open it in the PDF file browser. There is also to option of deleting the report with the Delete button.

Below is an example report from the Dashboard template generated and downloaded as a PDF file.
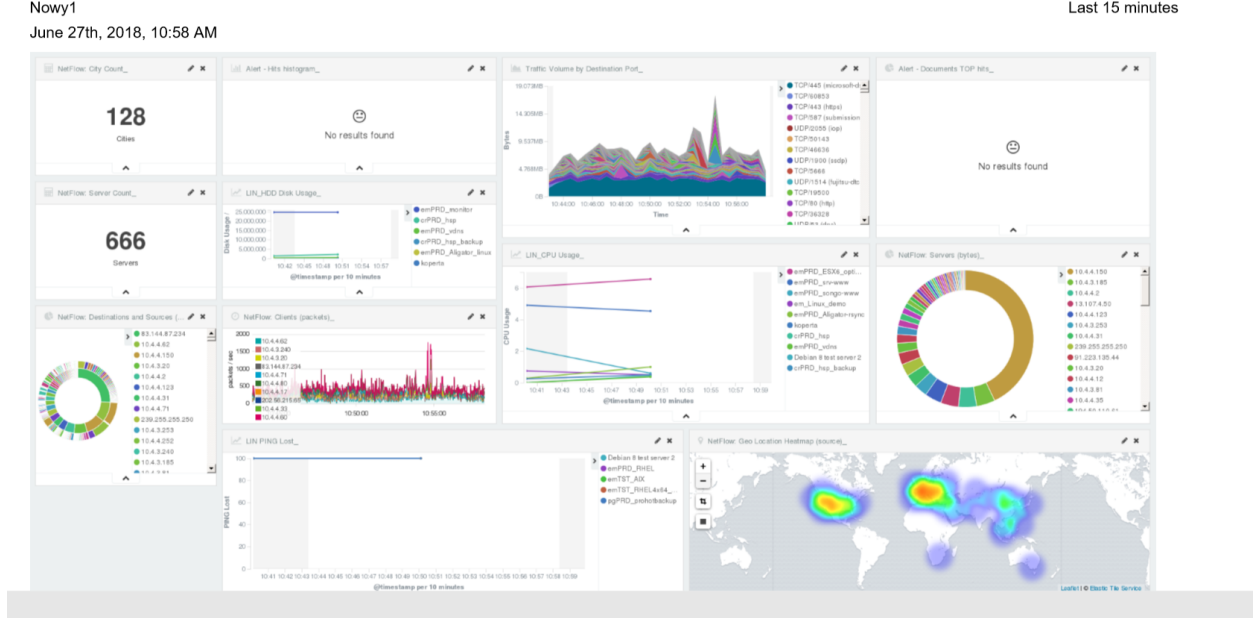
6.3 Scheduler Report (Schedule Export Dashboard)

In the Report selection, we have the option of setting the Scheduler which from Dashboard template can generate a report at time intervals. To do this goes to the Schedule Export Dashboard tab.

In this tab mark the saved Dashboard.

Data Export   Report Export   **Report Scheduler**

Create Schedule Task   Schedule Task List

Dashboard

Audit

[**Audit**] Dashboard

[AD] Removable Device **Audit**ing

[AD] File **Audit**

[AD] Servers **Audit**

[AD] Workstation **Audit**

Select Role

Cron Schedule

Daily

Submit

In the Email Topic field, enter the Message title, in the Email field enter the email address to which the report should be sent. From drop-down list choose at what frequency you want the report to be generated and sent. The action configured in this way is confirmed with the Submit button.

The defined action goes to the list and will generate a report to the e-mail address, with the cycle we set, until we cannot cancel it with the Cancel button.

User roles and object management

7.1 Users, roles and settings

ITRS Log Analytics allows to you manage users and permission for indexes and methods used by them. To do this click the "Config" button from the main menu bar.
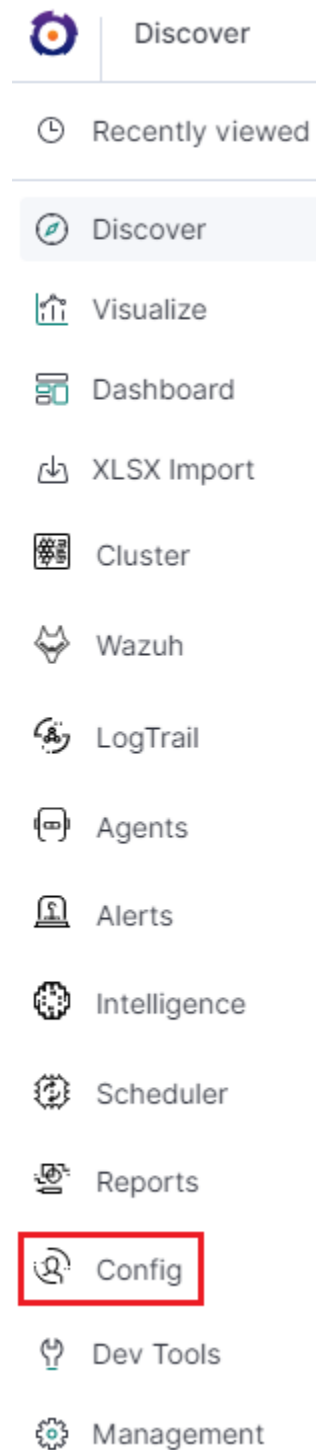
A new window will appear with three main tabs: „User Management", „Settings" and „License Info".

From the „User Management" level we have access to the following possibilities: Creating a user in „Create User", displaying users in „User List", creating new roles in „Create roles" and displaying existing roles in „List Role".

7.2 Creating a User (Create User)

7.2.1 Creating user

To create a new user click on the Config icon and you immediately enter the administration panel, where the first tab is to create a new user (**Create User**).

In the wizard that opens, we enter a unique username (Username field), password for the user (field Password) and assign a role (field Role). In this field we have the option of assigning more than one role. Until we select role in the Roles field, the Default Role field remains empty. When we mark several roles, these roles appear in the Default Role field. In this field we have the opportunity to indicate which role for a new user will be the default role with which the user will be associated in the first place when logging in. The default role field has one more important task - it binds all users with the field / role set in one group. When one of the users of this group create Visualization or Dashboard it will be available to other users from this role(group). Creating the account is confirmed with the Submit button.

7.2.2 User's modification and deletion, (User List)

Once we have created users, we can display their list. We do it in next tab (**User List**).

In this view, we get a list of user account with assigned roles and we have two buttons: Delete and Update. The first of these is ability to delete a user account. Under the Update button is a drop-down menu in which we can change the previous password to a new one (New password), change the password (Re-enter Ne Password), change the previously assigned roles (Roles), to other (we can take the role assigned earlier and give a new one, extend user permissions with new roles). The introduced changes are confirmed with the Submit button.

We can also see current user setting and clicking the Update button collapses the previously expanded menu.

7.3 Create, modify and delete a role (Create Role), (Role List)

In the Create Role tab we can define a new role with permissions that we assign to a pattern or several index patterns.

In example, we use the syslog2* index pattern. We give this name in the Paths field. We can provide one or more index patterns, their names should be separated by a comma. In the next Methods field, we select one or many methods that will be assigned to the role. Available methods:

- PUT - sends data to the server

- POST - sends a request to the server for a change

- DELETE - deletes the index / document

- GET - gets information about the index /document

- HEAD - is used to check if the index /document exists

In the role field, enter the unique name of the role. We confirm addition of a new role with the Submit button. To see if a new role has been added, go to the net Role List tab.

As we can see, the new role has been added to the list. With the Delete button we have the option of deleting it, while under the Update button we have a drop-down menu thanks to which we can add or remove an index pattern and add or remove a method. When we want to confirm the changes, we choose the Submit button. Pressing the Update button again will close the menu.

Fresh installation of the application have sewn solid roles which granting user special rights:

- admin - this role gives unlimited permissions to administer / manage the application

- alert - a role for users who want to see the Alert module

- kibana - a role for users who want to see the application GUI

- Intelligence - a role for users who are to see the Intelligence moduleObject access permissions (Objects permissions)

In the User Manager tab we can parameterize access to the newly created role as well as existing roles. In this tab we can indicate to which object in the application the role has access.

Example:

In the Role List tab we have a role called **sys2**, it refers to all index patterns beginning with syslog* and the methods get, post, delete, put and head are assigned.

When we go to the Object permission tab, we have the option to choose the sys2 role in the drop-down list choose a role:

After selecting, we can see that we already have access to the objects: two index patterns syslog2* and ITRS Log Analytics-* and on dashboard Windows Events. There are also appropriate read or updates permissions.

From the list we have the opportunity to choose another object that we can add to the role. We have the ability to quickly find this object in the search engine (Find) and narrowing the object class in the drop-down field "Select object type". The object type are associated with saved previously documents in the sections Dashboard, Index pattern, Search and Visualization. By buttons

we have the ability to add or remove or object, and Save button to save the selection.

7.4 Default user and passwords

The table below contains built-in user accounts and default passwords:

```
|Address                |User         |Password     |Role         |Description       ␣
↪                         |Usage        |
|-----------------------|-------------|-------------|-------------|------------------
↪-------------------------|---------------|
|https://localhost:5601       |logserver    |logserver    |logserver    |A built-in␣
↪*superuser* account                    |             |
|                       |alert        |alert        |alert        |A built-in account␣
↪for  the Alert module        |             |
|                       |intelligence |intelligece  |intelligence |A built-in account␣
↪for the Intelligence module  | authorizing communication with elasticsearch server |
|                       |scheduler    |scheduler    |scheduler    |A built-in account␣
↪for the Scheduler module     |
|                       |logstash     |logstash     |logstash     |A built-in account␣
↪for authorized comuunication form Logstash |
|                       |cerebro      |             |system acconut only     |A built-in␣
↪account for authorized comuunication from Cerebro moudule |
```

7.5 Changing password for the system account

After you change password for one of the system account (alert, intelligence, logserver, scheduler), you must to do appropriate changes in the application files.

1. Account **Logserver**

   • Update */etc/kibana/kibana.yml*:

```
vi /etc/kibana/kibana.yml
elasticsearch.password: new_logserver_passowrd
elastfilter.password: "new_logserver_password"
cerebro.password: "new_logserver_password"
```

2. Account **Intelligence**

   • Update */opt/ai/bin/conf.cfg*

```
vi /opt/ai/bin/conf.cfg
password=new_intelligence_password
```

3. Account **Alert**

   • Update file /opt/alert/config.yaml

```
vi /opt/alert/config.yaml
es_password: alert
```

4. Account **Scheduler**

   • Update */etc/kibana/kibana.yml*:

```
vi /etc/kibana/kibana.yml
elastscheduler.password: "new_scheduler_password"
```

5. Account **Logstash**

   • Update the Logstash pipeline configuration files (*.conf) in output sections:

```
vi /etc/logstash/conf.d/*.conf
elasticsearch {
        hosts => ["localhost:9200"]
        index => "syslog-%{+YYYY.MM}"
        user => "logstash"
        password => "new_password"
}
```

CHAPTER 8

Settings

8.1 General Settings

The Settings tab is used to set the audit on different activates or events and consists of several fields:

- **Time Out in minutes** field - this field defines the time after how many minutes the application will automatically log you off

- **Delete Application Tokens (in days)** - in this field we specify after what time the data from the audit should be deleted

- **Delete Audit Data (in days)** field - in this field we specify after what time the data from the audit should be deleted

- Next field are checkboxes in which we specify what kind of events are to be logged (saved) in the audit index. The events that can be monitored are: logging (Login), logging out (Logout), creating a user (Create User), deleting a user (Delete User), updating user (Update User), creating a role (Create Role), deleting a role (Delete Role), update of the role (Update Role), start of export (Export Start), delete of export (Export Delete), queries (Queries), result of the query (Content), if attempt was made to perform a series of operation (Bulk)

- **Delete Exported CSVs (in days)** field - in this field we specify after which time exported file with CSV extension have to be removed

- **Delete Exported PDFs (in days)** field - in this field we specify after which time exported file with PDF extension have to be removed

To each field is assigned "Submit" button thanks to which we can confirm the changes.

8.2 License (License Info)

The License Information tab consists of several non-editable information fields.

User Management   Settings   License Info

Company: Foo Bar

Data nodes in cluster : 10

No of documents :

Indices : [*]

Issued on : 2019-05-30T08:49:20.042034300

Validity : 120 months

Version : 7.0.1

These fields contain information:

- Company field, who owns the license - in this case EMCA S.A.

- Data nodes in cluster field - how many nodes we can put in one cluster - in this case 100

- No of documents field - empty field

- Indices field - number of indexes, symbol[*] means that we can create any number of indices

- Issued on field - date of issue

- Validity field - validity, in this case for 360000 months

8.3 Special accounts

At the first installation of the ITRS Log Analytics application, apart from the administrative account (logserver), special applications are created in the application: alert, intelligence and scheduler.

- **Alert Account** - this account is connected to the Alert Module which is designed to track events written to the index for the previously defined parameters. If these are met the information action is started (more on the action in the Alert section)

- **Intelligence Account** - with this account is related to the module of artificial intelligence which is designed to track events and learn the network based on previously defined rules artificial intelligence based on one of the available algorithms (more on operation in the Intelligence chapter)

- **Scheduler Account** - the scheduler module is associated with this account, which corresponds to, among others for generating reports

Alert Module

ITRS Log Analytics allows you to create alerts, i.e. monitoring queries. These are constant queries that run in the background and when the conditions specified in the alert are met, the specify action is taken.

Create alert rule	Alert rules List	Alerts Status	Playbook	Risks	Incidents

For example, if you want to know when more than 20 „status:500" responscode from on our homepage appear within an one hour, then we create an alert that check the number of occurrences of the „status:500" query for a specific index every 5 minutes. If the condition we are interested in is met, we send an action in the form of sending a message to our e-mail address. In the action, you can also set the launch of any script.

9.1 Enabling the Alert Module

To enabling the alert module you should:

- generate writeback index for Alert service:

Only applies to versions 6.1.5 and older. From version 6.1.6 and later, the Alert index is created automatically

```
/opt/alert/bin/elastalert-create-index --config /opt/alert/config.yaml
```

- configure the index pattern for alert*

- start the Alert service:

```
systemctl start alert
```

9.2 Creating Alerts

To create the alert, click the "Alerts" button from the main menu bar.
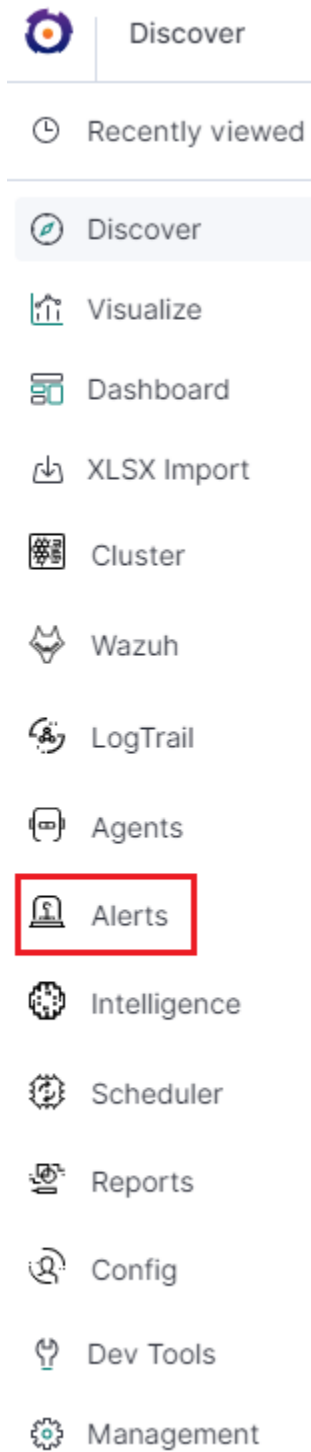
We will display a page with tree tabs: Create new alerts in „Create alert rule", manage alerts in „Alert rules List" and check alert status „Alert Status".

In the alert creation windows we have an alert creation form:

| Create alert rule | Alert rules List | Alerts Status | Playbook | Risks | Incidents |

Create Alert

Name

Alert Rule Name

Index pattern

Index pattern

Read fields

Risk key

Aggregation type

max ▼

Rule importance (1 - 100%)

50

Role

admin
alert
intelligence
kibana

Type

▼

Description

Example

Example

Alert method

None ▼

Any

Playbooks

Test rule

- **Name** - the name of the alert, after which we will recognize and search for it.
- **Index pattern** - a pattern of indexes after which the alert will be searched.
- **Role** - the role of the user for whom an alert will be available
- **Type** - type of alert
- **Description** - description of the alert.
- **Example** - an example of using a given type of alert. Descriptive field
- **Alert method** - the action the alert will take if the conditions are met (sending an email message or executing a command)
- **Any** - additional descriptive field.# List of Alert rules #

The "Alert Rule List" tab contain complete list of previously created alert rules:

In this window, you can activate / deactivate, delete and update alerts by clicking on the selected icon with the given

alert: .

9.3 Alerts status

In the "Alert status" tab, you can check the current alert status: if it activated, when it started and when it ended, how long it lasted, how many event sit found and how many times it worked.

Also, on this tab, you can recover the alert dashboard, by clicking the "Recovery Alert Dashboard" button.# Type of the Alert module rules #

The various RuleType classes, defined in ITRS Log Analytics-Log-Aalytics. An instance is held in memory for each rule, passed all of the data returned by querying Elasticsearch with a given filter, and generates matches based on that data.

- *Any* - The any rule will match everything. Every hit that the query returns will generate an alert.

- *Blacklist* - The blacklist rule will check a certain field against a blacklist, and match if it is in the blacklist.
- *Whitelist* - Similar to blacklist, this rule will compare a certain field to a whitelist, and match if the list does not contain the term.
- *Change* - This rule will monitor a certain field and match if that field changes.
- *Frequency* - his rule matches when there are at least a certain number of events in a given time frame.
- *Spike* - This rule matches when the volume of events during a given time period is spike_height times larger or smaller than during the previous time period.
- *Flatline* - This rule matches when the total number of events is under a given threshold for a time period.
- *New Term* - This rule matches when a new value appears in a field that has never been seen before.
- *Cardinality* - This rule matches when a the total number of unique values for a certain field within a time frame is higher or lower than a threshold.
- *Metric Aggregation* - This rule matches when the value of a metric within the calculation window is higher or lower than a threshold.
- *Percentage Match* - This rule matches when the percentage of document in the match bucket within a calculation window is higher or lower than a threshold.
- *Unique Long Term* - This rule matches when there are values of compare_key in each checked timeframe.
- *Find Match* - Rule match when in defined period of time, two correlated documents match certain strings.
- *Difference* - Rule matches for value difference between two aggregations calculated for different periods in time.
- *ConsecutiveGrowth* - Rule matches for value difference between two aggregations calculated for different periods in time.

9.4 Example of rules

9.4.1 Unix - Authentication Fail

- index pattern:

```
syslog-*
```

- Type:

```
Frequency
```

- Alert Method:

```
Email
```

- Any:

```
num_events: 4
timeframe:
  minutes: 5

filter:
```

<div align="right">(continues on next page)</div>

```
    - query_string:
        query: "program: (ssh OR sshd OR su OR sudo) AND message: \"Failed password\
↪""
```

9.4.2 Windows - Firewall disable or modify

- index pattern:

```
beats-*
```

- Type:

```
Any
```

- Alert Method:

```
Email
```

- Any:

filter:

```
      - query_string:
            query: "event_id:(4947 OR 4948 OR 4946 OR 4949 OR 4954 OR 4956 OR 5025)
↪"
```

9.4.3 SIEM Rules

Beginning with version 6.1.7, the following SIEM rules are delivered with the product.

```
| Nr. | Architecture/Application | Rule Name
↪         | Description
↪
↪
↪
↪
↪
↪
↪                                                                   | Index name  ␣
↪| Requirements                                           | Source                |␣
↪Time definition           | Threshold |
|-----|--------------------------|--------------------------------------------------
↪----------|----------------------------------------------------------------------
↪----------------------------------------------------------------------------------
↪----------------------------------------------------------------------------------
↪----------------------------------------------------------------------------------
↪----------------------------------------------------------------------------------
↪----------------------------------------------------------------------------------
↪----------------------------------------------------------------------------------
↪-------------------------------------------------------------|--------------
↪|--------------------------------------------------|-----------------------------|-
↪---------------------------|-----------|
| 1   | Windows                  | Windows - Admin night logon
↪         | Alert on Windows login events when detected outside business hours    ␣
↪
```

```
↪
↪
↪                                                                   | winlogbeat-*␣
```

```
| 2   | Windows                   | Windows - Admin task as user          ␣
→        | Alert when admin task is initiated by regular user.  Windows event id␣
→4732 is verified towards static admin list. If the user does not belong to admin␣
→list AND the event is seen than we generate alert. Static Admin list is a logstash ␣
→disctionary file that needs to be created manually. During Logstash lookup a field␣
→user.role:admin is added to an event.4732: A member was added to a security-enabled␣
→local group                                                                      ␣
→                                                                                  ␣
→                                                                                |␣
→winlogbeat-* | winlogbeatLogstash admin dicstionary lookup file | Widnows Security␣
→Eventlog    | Every 1min                   | 1         |
| 3   | Windows                   | Windows - diff IPs logon               ␣
→        | Alert when Windows logon process is detected and two or more different␣
→IP addressed are seen in source field.  Timeframe is last 15min.Detection is based␣
→onevents 4624 or 1200.4624: An account was successfully logged on1200: Application␣
→token success                                                                     ␣
→                                                                                  ␣
→                                                                                  ␣
→                                                                                  ␣
→                                                          | winlogbeat-
→* | winlogbeat                              | Widnows Security Eventlog    ␣
→| Every 1min, for last 15min | 1           |
| 4   | Windows                   | Windows - Event service error         ␣
→        | Alert when Windows event 1108 is matched1108: The event logging service␣
→encountered an error                                                              ␣
→                                                                                  ␣
→                                                                                  ␣
→                                                                                  ␣
→                                                                                  ␣
→                                                                                  ␣
→                                                          | winlogbeat-*␣
→| winlogbeat                              | Widnows Security Eventlog    |␣
→Every 1min                   | 1         |
| 5   | Windows                   | Windows - file insufficient privileges  ␣
→        | Alert when Windows event 5145 is matched5145: A network share object␣
→was checked to see whether client can be granted desired accessEvery time a network␣
→share object (file or folder) is accessed, event 5145 is logged. If the access is␣
→denied at the file share level, it is audited as a failure event. Otherwise, it␣
→considered a success. This event is not generated for NTFS access.                 ␣
→                                                                                  ␣
→                                                                                  ␣
→                                                                                |␣
→winlogbeat-* | winlogbeat                              | Widnows Security␣
→Eventlog    | Every 1min, for last 15min | 50          |
| 6   | Windows                   | Windows - Kerberos pre-authentication failed  ␣
→        | Alert when Windows event 4625 or 4771 is matched 4625: An account␣
→failed to log on 4771: Kerberos pre-authentication failed                         ␣
→                                                                                  ␣
→                                                                                  ␣
→                                                                                  ␣
→                                                                                  ␣
→                                                                                  ␣
→                                                                                |␣
→winlogbeat-* | winlogbeat                              | Widnows Security␣
→Eventlog    | Every 1min                   | 1         |
| 7   | Windows                   | Windows - Logs deleted                ␣
→        | Alert when Windows event 1102 OR 104 is matched1102: The audit log was␣
→cleared104: Event log cleared
→                                                                                  ␣
→
```

```
→                                                                                  ␣
→                                                                                  ␣
→                                                          | winlogbeat-*␣
```

```
| 8    | Windows                    | Windows - Member added to a security-enabled␣
→global group    | Alert when Windows event 4728 is matched4728: A member was added␣
→to a security-enabled global group
→                                                                                  ␣
→                                                                                  ␣
→                                                                                  ␣
→                                                                                  ␣
→                                                                                  ␣
→                                                                                  |␣
→winlogbeat-* | winlogbeat                                      | Widnows Security␣
→Eventlog    | Every 1min              | 1          |
| 9    | Windows                    | Windows - Member added to a security-enabled local␣
→group     | Alert when Windows event 4732 is matched4732: A member was added to a␣
→security-enabled local group
→                                                                                  ␣
→                                                                                  ␣
→                                                                                  ␣
→                                                                                  ␣
→                                                                                  ␣
→                                                                | winlogbeat-
→* | winlogbeat                                      | Widnows Security Eventlog   ␣
→| Every 1min              | 1          |
| 10   | Windows                    | Windows - Member added to a security-enabled␣
→universal group | Alert when Windows event 4756 is matched4756: A member was added␣
→to a security-enabled universal group
→                                                                                  ␣
→                                                                                  ␣
→                                                                                  ␣
→                                                                                  ␣
→                                                                                  ␣
→                                                                                  |␣
→winlogbeat-* | winlogbeat                                      | Widnows Security␣
→Eventlog    | Every 1min              | 1          |
| 11   | Windows                    | Windows - New device␣
→           | Alert when Windows event 6414 is matched6416: A new external device was␣
→recognized by the system
→                                                                                  ␣
→                                                                                  ␣
→                                                                                  ␣
→                                                                                  ␣
→                                                                                  ␣
→                                                                | winlogbeat-*␣
→| winlogbeat                                      | Widnows Security Eventlog    |␣
→Every 1min              | 1          |
| 12   | Windows                    | Windows - Package installation␣
→           | Alert when Windows event 4697 is matched 4697: A service was installed␣
→in the system
→                                                                                  ␣
→                                                                                  ␣
→                                                                                  ␣
→                                                                                  ␣
→                                                                                  ␣
→                                                                | winlogbeat-*␣
→| winlogbeat                                      | Widnows Security Eventlog    |␣
→Every 1min              | 1          |
| 13   | Windows                    | Windows - Password policy change␣
→           | Alert when Windows event 4739 is matched4739: Domain Policy was changed␣
→
→
→
```

9.4. Example of rules

81

```
→                                                                                  ␣
→                                                                                  ␣
→                                                                | winlogbeat-*␣
```

```
| 14  | Windows                 | Windows - Security log full                      ␣
→       | Alert when Windows event 1104 is matched1104: The security Log is now␣
→full                                                                             ␣
→                                                                                 ␣
→                                                                                 ␣
→                                                                                 ␣
→                                                                                 ␣
→                                                                                 ␣
→                                                        | winlogbeat-
→* | winlogbeat                             | Widnows Security Eventlog   ␣
→| Every 1min                | 1          |
| 15  | Windows                 | Windows - Start up                               ␣
→       | Alert when Windows event 4608 is matched 4608: Windows is starting up ␣
→                                                                                 ␣
→                                                                                 ␣
→                                                                                 ␣
→                                                                                 ␣
→                                                                                 ␣
→                                                                                 ␣
→                                                        | winlogbeat-*␣
→| winlogbeat                             | Widnows Security Eventlog   |␣
→Every 1min                | 1          |
| 16  | Windows                 | Windows - Account lock                           ␣
→       | Alert when Windows event 4740 is matched4740: A User account was Locked␣
→out                                                                              ␣
→                                                                                 ␣
→                                                                                 ␣
→                                                                                 ␣
→                                                                                 ␣
→                                                                                 ␣
→                                                        | winlogbeat-*␣
→| winlogbeat                             | Widnows Security Eventlog   |␣
→Every 1min                | 1          |
| 17  | Windows                 | Windows - Security local group was changed       ␣
→       | Alert when Windows event 4735 is matched4735: A security-enabled local␣
→group was changed                                                                ␣
→                                                                                 ␣
→                                                                                 ␣
→                                                                                 ␣
→                                                                                 ␣
→                                                                                 ␣
→                                                        | winlogbeat-*␣
→| winlogbeat                             | Widnows Security Eventlog   |␣
→Every 1min                | 1          |
| 18  | Windows                 | Windows - Reset password attempt                 ␣
→       | Alert when Windows event 4724 is matched4724: An attempt was made to␣
→reset an accounts password                                                       ␣
→                                                                                 ␣
→                                                                                 ␣
→                                                                                 ␣
→                                                                                 ␣
→                                                                                 ␣
→                                                        | winlogbeat-
→* | winlogbeat                             | Widnows Security Eventlog   ␣
→| Every 1min                | 1          |
| 19  | Windows                 | Windows - Code integrity changed                 ␣
→       | Alert when Windows event 5038 is matched5038: Detected an invalid image␣
→hash of a fileInformation:   Code Integrity is a feature that improves the␣
→security of the operating system by validating the integrity of a driver or system␣
→file each time it is loaded into memory.   Code Integrity detects whether an␣
→unsigned driver or system file is being loaded into the kernel, or whether a system␣
→file has been modified by malicious software that is being run by a user account␣
→with administrative permissions.   On x64-based versions of the operating system,␣
→kernel-mode drivers must be digitally signed.The event logs the following␣
```

Chapter 9. Alert Module

```
| 20  | Windows                  | Windows - Application error                            ␣
→          | Alert when Windows event 1000 is matched1000: Application error        ␣
→                                                                                   ␣
→                                                                                   ␣
→                                                                                   ␣
→                                                                                   ␣
→                                                                                   ␣
→                                                                                   ␣
→                                                                 | winlogbeat-*␣
→| winlogbeat                                        | Widnows Application Eventlog |␣
→Every 1min                | 1          |
| 21  | Windows                  | Windows - Application hang                             ␣
→          | Alert when Windows event 1001 OR 1002 is matched1001: Application fault␣
→bucket1002: Application hang                                                        ␣
→                                                                                   ␣
→                                                                                   ␣
→                                                                                   ␣
→                                                                                   ␣
→                                                                                   ␣
→                                                                 | winlogbeat-*␣
→| winlogbeat                                        | Widnows Application Eventlog |␣
→Every 1min                | 1          |
| 22  | Windows                  | Windows - Audit policy changed                        ␣
→          | Alert when Windows event 4719 is matched4719: System audit policy was␣
→changed                                                                            ␣
→                                                                                   ␣
→                                                                                   ␣
→                                                                                   ␣
→                                                                                   ␣
→                                                                                   ␣
→                                                                 | winlogbeat-
→* | winlogbeat                                        | Widnows Security Eventlog   ␣
→| Every 1min                | 1          |
| 23  | Windows                  | Windows - Eventlog service stopped                    ␣
→          | Alert when Windows event 6005 is matched6005: Eventlog service stopped ␣
→                                                                                   ␣
→                                                                                   ␣
→                                                                                   ␣
→                                                                                   ␣
→                                                                                   ␣
→                                                                                   ␣
→                                                                 | winlogbeat-*␣
→| winlogbeat                                        | Widnows Security Eventlog   |␣
→Every 1min                | 1          |
| 24  | Windows                  | Windows - New service installed                       ␣
→          | Alert when Windows event 7045 OR 4697 is matched7045,4697: A service␣
→was installed in the system                                                        ␣
→                                                                                   ␣
→                                                                                   ␣
→                                                                                   ␣
→                                                                                   ␣
→                                                                                   ␣
→                                                                 | winlogbeat-
→* | winlogbeat                                        | Widnows Security Eventlog   ␣
→| Every 1min                | 1          |
| 25  | Windows                  | Windows - Driver loaded                                ␣
→          | Alert when Windows event 6 is matched6: Driver loadedThe driver loaded␣
→events provides information about a driver being loaded on the system. The␣
→configured hashes are provided as well its signature information. The signature is␣
→created asynchronously for performance reasons and indicates if the file was␣
→removed after loading.                                                             ␣
→                                                                                   ␣
→                                                                                   ␣
→                                                                                   ␣
```

9.4. Example of rules

83

```
| 26  | Windows                 | Windows - Firewall rule modified              ␣
→         | Alert when Windows event 2005 is matched2005: A Rule has been modified␣
→in the Windows firewall Exception List                                         ␣
→                                                                               ␣
→                                                                               ␣
→                                                                               ␣
→                                                                               ␣
→                                                                               ␣
→                                                    | winlogbeat-*␣
→| winlogbeat                                    | Widnows Security Eventlog    |␣
→Every 1min                | 1          |
| 27  | Windows                 | Windows - Firewall rule add                   ␣
→         | Alert when Windows event 2004 is matched2004: A firewall rule has been␣
→added                                                                          ␣
→                                                                               ␣
→                                                                               ␣
→                                                                               ␣
→                                                                               ␣
→                                                                               ␣
→                                                    | winlogbeat-*␣
→| winlogbeat                                    | Widnows Security Eventlog    |␣
→Every 1min                | 1          |
| 28  |                         | Windows - Firewall rule deleted               ␣
→         | Alert when Windows event 2006 or 2033 or 2009 is matched2006,2033,2009:␣
→Firewall rule deleted                                                          ␣
→                                                                               ␣
→                                                                               ␣
→                                                                               ␣
→                                                                               ␣
→                                                                               ␣
→                                                    | winlogbeat-*␣
→| winlogbeat                                    | Widnows Security Eventlog    |␣
→Every 1min                | 1          |
```

9.5 Playbooks

ITRS Log Analytics has a set of predefined set of rules and activities (called Playbook) that can be attached to a registered event in the Alert module. Playbooks can be enriched with scripts that can be launched together with Playbook.

9.5.1 Create Playbook

To add a new playbook, go to the **Alert** module, select the **Playbook** tab and then **Create Playbook**

In the **Name** field, enter the name of the new Playbook.

In the **Text** field, enter the content of the Playbook message.

In the **Script** field, enter the commands to be executed in the script.

To save the entered content, confirm with the **Submit** button.

9.5.2 Playbooks list

To view saved Playbook, go to the **Alert** module, select the **Playbook** tab and then **Playbooks list**:
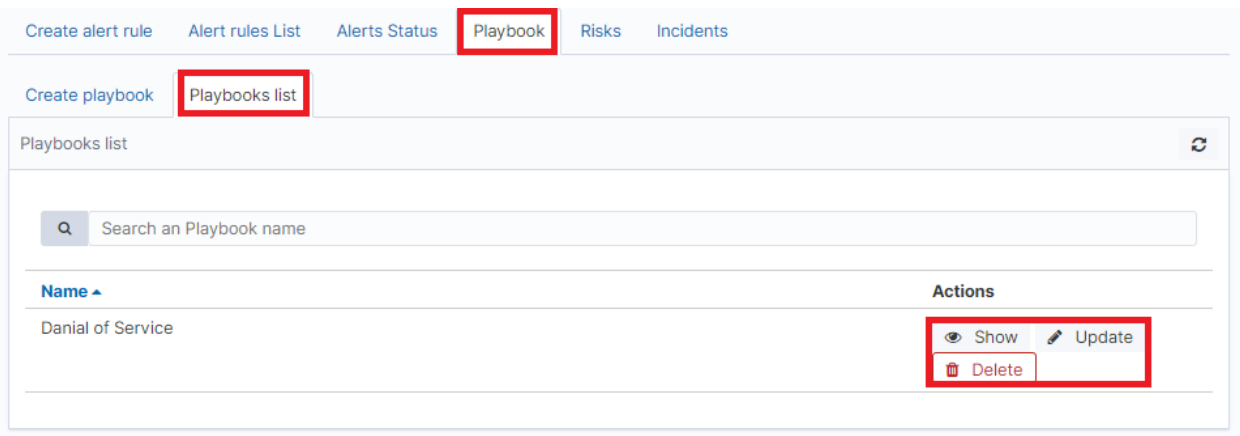
To view the content of a given Playbook, select the **Show** button.

To enter the changes in a given Playbook or in its script, select the **Update** button. After making changes, select the **Submit** button.

To delete the selected Playbook, select the **Delete** button.

9.5.3 Linking Playbooks with alert rule

You can add a Playbook to the Alert while creating a new Alert or by editing a previously created Alert.

To add Palybook to the new Alert rule, go to the **Create alert rule** tab and in the **Playbooks** section use the arrow keys to move the correct Playbook to the right window.

To add a Palybook to existing Alert rule, go to the **Alert rule list** tab with the correct rule select the **Update** button and in the **Playbooks** section use the arrow keys to move the correct Playbook to the right window.

9.5.4 Playbook verification

When creating an alert or while editing an existing alert, it is possible that the system will indicate the most-suited playbook for the alert. For this purpose, the Validate button is used, which starts the process of searching the existing playbook and selects the most appropriate ones.

9.6 Risks

ITRS Log Analytics allows you to estimate the risk based on the collected data. The risk is estimated based on the defined category to which the values from 0 to 100 are assigned.

Information on the defined risk for a given field is passed with an alert and multiplied by the value of the Rule Importance parameter.

9.6.1 Create category

To add a new risk Category, go to the **Alert** module, select the **Risks** tab and then **Create Cagtegory**.
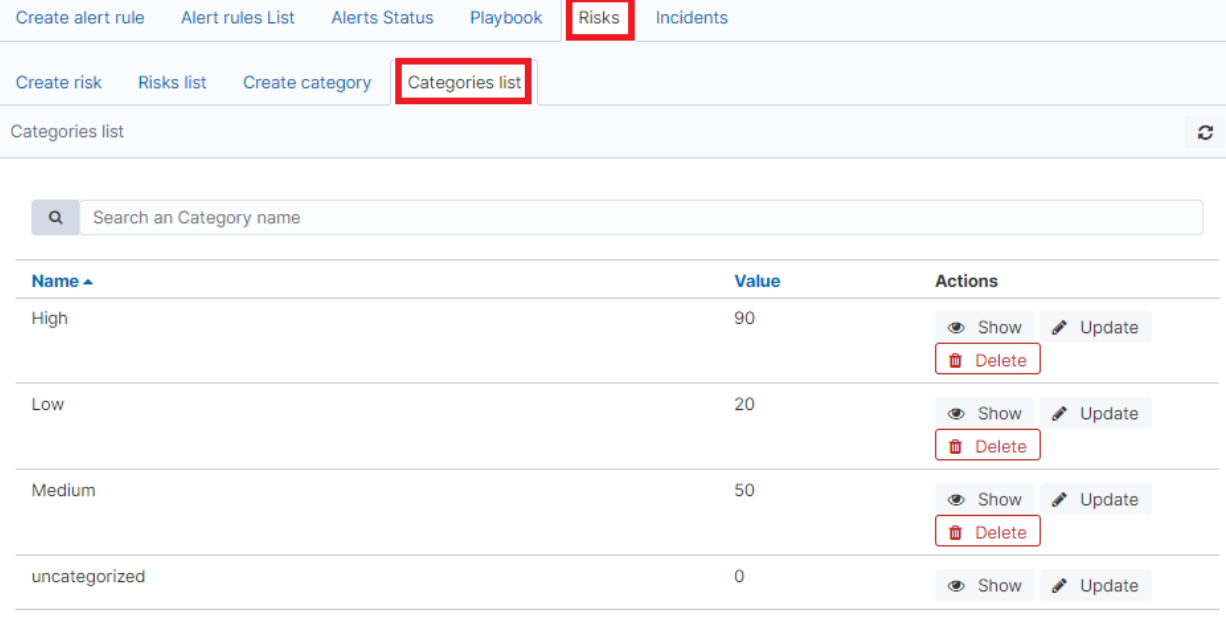
Enter the **Name** for the new category and the category **Value**.

9.6.2 Category list

To view saved Category, go to the **Alert** module, select the **Risks** tab and then **Categories list**:

To view the content of a given Category, select the **Show** button.

To change the value assigned to a category, select the **Update** button. After making changes, select the **Submit** button.

To delete the selected Category, select the **Delete** button.

9.6.3 Create risk

To add a new playbook, go to the Alert module, select the Playbook tab and then Create Playbook

In the **Index pattern** field, enter the name of the index pattern. Select the **Read fields** button to get a list of fields from the index. From the box below, select the field name for which the risk will be determined.

From the **Timerange field**, select the time range from which the data will be analyzed.

Press the **Read valules** button to get values from the previously selected field for analysis.

Next, you must assign a risk category to the displayed values. You can do this for each value individually or use the check-box on the left to mark several values and set the category globally using the **Set global category** button. To quickly find the right value, you can use the search field.

After completing, save the changes with the **Submit** button.

9.6.4 List risk

To view saved risks, go to the **Alert** module, select the **Risks** tab and then **Risks list**:

To view the content of a given Risk, select the **Show** button.

To enter the changes in a given Risk, select the **Update** button. After making changes, select the **Submit** button.

To delete the selected Risk, select the **Delete** button.

9.6.5 Linking risk with alert rule

You can add a Risk key to the Alert while creating a new Alert or by editing a previously created Alert.

To add Risk key to the new Alert rule, go to the **Create alert rule** tab and after entering the index name, select the **Read fields** button and in the **Risk key** field, select the appropriate field name. In addition, you can enter the validity of the rule in the **Rule Importance** field (in the range 1-100%), by which the risk will be multiplied.

To add Risk key to the existing Alert rule, go to the **Alert rule list**, tab with the correct rule select the **Update** button. Use the **Read fields** button and in the **Risk key** field, select the appropriate field name. In addition, you can enter the validity of the rule in the **Rule Importance**.

9.6.6 Risk calculation algorithms

The risk calculation mechanism performs the aggregation of the risk field values. We have the following algorithms for calculating the alert risk (Aggregation type):

- min - returns the minimum value of the risk values from selected fields;
- max - returns the maximum value of the risk values from selected fields;
- avg - returns the average of risk values from selected fields;
- sum - returns the sum of risk values from selected fields;
- custom - returns the risk value based on your own algorithm

9.6.7 Adding a new risk calculation algorithm

The new algorithm should be added in the `./elastalert_modules/playbook_util.py` file in the `calculate_risk` method. There is a sequence of conditional statements for already defined algorithms:

```python
#aggregate values by risk_key_aggregation for rule
if risk_key_aggregation == "MIN":
    value_agg = min(values)
elif risk_key_aggregation == "MAX":
    value_agg = max(values)
elif risk_key_aggregation == "SUM":
    value_agg = sum(values)
elif risk_key_aggregation == "AVG":
    value_agg = sum(values)/len(values)
else:
    value_agg = max(values)
```

To add a new algorithm, add a new sequence as shown in the above code:

```python
elif risk_key_aggregation == "AVG":
    value_agg = sum(values)/len(values)
elif risk_key_aggregation == "AAA":
    value_agg = BBB
else:
    value_agg = max(values)
```

where **AAA** is the algorithm code, **BBB** is a risk calculation function.

9.6.8 Using the new algorithm

After adding a new algorithm, it is available in the GUI in the Alert tab.

To use it, add a new rule according to the following steps:

- Select the `custom` value in the `Aggregation` type field;
- Enter the appropriate value in the `Any` field, e.g. `risk_key_aggregation:  AAA`

The following figure shows the places where you can call your own algorithm:

9.6.9 Additional modification of the algorithm (weight)

Below is the code in the `calcuate_risk` method where category values are retrieved - here you can add your weight:

```python
        #start loop by tablicy risk_key
        for k in range(len(risk_keys)):
            risk_key = risk_keys[k]
            logging.info(' >>>>>>>>>>>>>> risk_key: ')
            logging.info(risk_key)
            key_value = lookup_es_key(match, risk_key)
            logging.info(' >>>>>>>>>>>>>> key_value: ')
            logging.info(key_value)
            value = float(self.get_risk_category_value(risk_key, key_value))
            values.append( value )
            logging.info(' >>>>>>>>>>>>>> risk_key values: ')
            logging.info(values)
        #finish loop by tablicy risk_key
        #aggregate values by risk_key_aggregation form rule
        if risk_key_aggregation == "MIN":
```

(continues on next page)

```
            value_agg = min(values)
        elif risk_key_aggregation == "MAX":
            value_agg = max(values)
        elif risk_key_aggregation == "SUM":
            value_agg = sum(values)
        elif risk_key_aggregation == "AVG":
            value_agg = sum(values)/len(values)
        else:
            value_agg = max(values)
```

`Risk_key` is the array of selected risk key fields in the GUI. A loop is made on this array and a value is collected for the categories in the line:

```
value = float(self.get_risk_category_value(risk_key, key_value))
```

Based on, for example, Risk_key, you can multiply the value of the value field by the appropriate weight. The value field value is then added to the table on which the risk calculation algorithms are executed.

9.7 Indicators of compromise (IoC)

ITRS Log-Analytics has the Indicators of compromise (IoC) functionality, which is based on the Malware Information Sharing Platform (MISP). IoC observes the logs sent to the system and marks documents if their content is in MISP signature. Based on IoC markings, you can build alert rules or track incident behavior.

9.7.1 Configuration

Bad IP list update

To update bad reputation lists and to create `.blacklists` index, you have to run following scripts:

```
/etc/logstash/lists/bin/badreputation_iplists.sh
/etc/logstash/lists/bin/misp_threat_lists.sh
```

Scheduling bad IP lists update

This can be done in `cron` (host with Logstash installed):

```
0 1 * * * logstash /etc/logstash/lists/bin/badreputation_iplists.sh
0 6 * * * logstash /etc/logstash/lists/bin/misp_threat_lists.sh
```

or with Kibana Scheduller app (**only if Logstash is running on the same host**).

- Prepare script path:

```
/bin/ln -sfn /etc/logstash/lists/bin /opt/ai/bin/lists
chown logstash:kibana /etc/logstash/lists/
chmod g+w /etc/logstash/lists/
```

- Log in to Energy Logserver GUI and go to **Scheduler** app. Set it up with below options and push "Submit" button:

```
Name:           BadReputationList
Cron pattern:   0 1 * * *
Command:        lists/badreputation_iplists.sh
Category:       logstash
```

and second:

```
Name:           MispThreatList
Cron pattern:   0 1 * * *
Command:        lists/misp_threat_lists.sh
Category:       l\logstash
```

After a couple of minutes check for blacklists index:

```
curl -sS -u user:password -XGET '127.0.0.1:9200/_cat/indices/.blacklists?s=index&v'
health status index         uuid                   pri rep docs.count docs.deleted␣
→store.size pri.store.size
green  open   .blacklists Mld2Qe2bSRuk2VyKm-KoGg   1   0     76549           0     ␣
→4.7mb          4.7mb
```

Configuration alert rule

CHAPTER 10

Intelligence Module

A dedicated artificial intelligence module has been built in the ITRS Log Analytics system that allows prediction of parameter values relevant to the maintenance of infrastructure and IT systems. Such parameters include:

- use of disk resources,
- use of network resources,
- using the power of processors
- detection of known incorrect behaviour of IT systems

To access of the Intelligence module, click the tile icon from the main meu bar and then go to the „Intelligence" icon (To go back, click to the „Search" icon).
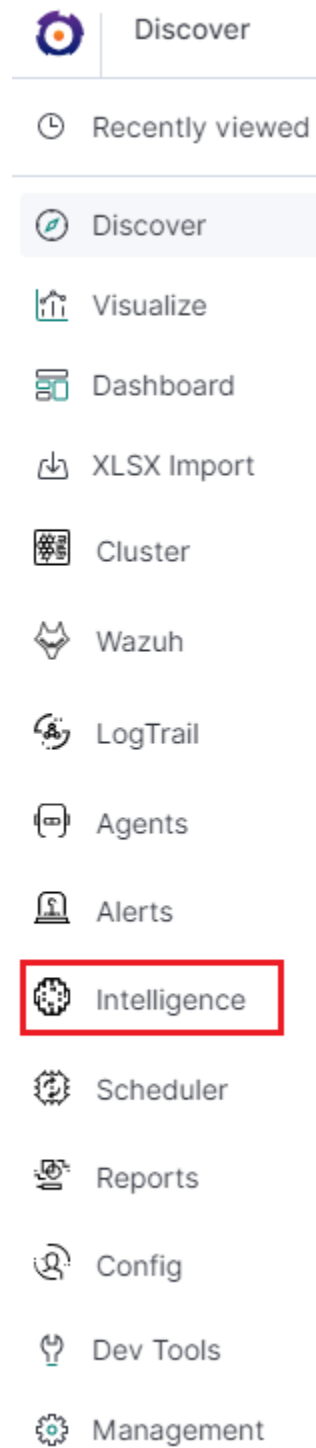
Create AI Rule | AI Rules List | AI Learn | AI Learn Tasks

There are 4 screens available in the module:

- **Create AI Rule** - the screen allows you to create artificial intelligence rules and run them in scheduler mode or immediately

- **AI Rules List** - the screen presents a list of created artificial intelligence rules with the option of editing, previewing and deleting them

- **AI Learn** - the screen allows to define the conditions for teaching the MLP neural network

- **AI Learn Tasks** - a screen on which the initiated and completed learning processes of neural networks with the ability to preview learning results are presented.# Create AI Rule #

To create the AI Rule, click on the tile icon from the main menu bar, go to the „Intelligence" icon and select "Create AI Rule" tab. The screen allows to defining the rules of artificial intelligence based on one of the available algorithms (a detailed description of the available algorithms is available in a separate document).

Description of the controls available on the fixed part of screen:

- **Algorithm** - the name of the algorithm that forms the basis of the artificial intelligence rule

- **Choose search** - search defined in the ITRS Log Analytics system, which is used to select a set of data on which the artificial intelligence rule will operate

- **Run** - a button that allows running the defined AI rule or saving it to the scheduler and run as planned

The rest of the screen will depend on the chosen artificial intelligence algorithm.

10.1 The fixed part of the screen

Description of the controls available on the fixed part of screen:

- Algorithm - the name of the algorithm that forms the basis of the artificial intelligence rule

- Choose search - search defined in the ITRS Log Analytics system, which is used to select a set of data on which the artificial intelligence rule will operate

- Run - a button that allows running the defined AI rule or saving it to the scheduler and run as planned

The rest of the screen will depend on the chosen artificial intelligence algorithm.

10.2 Screen content for regressive algorithms

Algorithm:

Simple Moving Average

Choose search:

Uslugi_WWW_with_cols

AI Rule Name:	my_test_
Feature to analyse (from search):	perf_data./
Multiply by field (from search):	hostname
Multiply by values (from search):	emPRD_Aligator_linux emPRD_Cyberoam_public_FC emPRD_ESX6_optima64 emPRD_RHEL
Time frame:	Day
Value type:	Average
Max probes:	20
Max predictions:	30
Data limit:	10000000
Start date:	2018-04-06 09:51:31
Scheduler:	
Role:	admin ALL_test audit

Description of controls:

- **feature to analyze from search** - analyzed feature (dictated)

- **multiply by field** - enable multiplication of algorithms after unique values of the feature indicated here. Multiplication allows you to run the AI rule one for e.g. all servers. The value "none" in this field means no multiplication.

- **multiply by values** - if a trait is indicated in the „multiply by field", then unique values of this trait will appear in this field. Multiplications will be made for the selected values. If at least one of value is not selected, the „Run" buttons will be inactive.'

In other words, multiplication means performing an analysis for many values from the indicated field, for example: `sourece_node_host`- which we indicate in `Multiply by field (from search)`.

However, in `Multiply by values (from search)` we already indicate values of this field for which the analysis will be performed, for example: host1, host2, host3, ...

- **time frame** - feature aggregation method (1 minute, 5 minute, 15 minute, 30 minute, hourly, weekly, monthly, 6 months, 12 months)

- **max probes** - how many samples back will be taken into account for analysis. A single sample is an aggregated data according to the aggregation method.

- **value type** - which values to take into account when aggregating for a given time frame (e.g. maximum from time frame, minimum, average)

- **max predictions** - how many estimates we make for ahead (we take time frame)

- **data limit** - limits the amount of date downloaded from the source. It speeds up processing but reduces its quality

- **start date** - you can set a date earlier than the current date in order to verify how the selected algorithm would work on historical data

- **Scheduler** - a tag if the rule should be run according to the plan for the scheduler. If selected, additional fields will appear;

- **Prediction cycle** - plan definition for the scheduler, i.e. the cycle in which the prediction rule is run (e.g. once a day, every hour, once a week). In the field, enter the command that complies with the cron standard. Enable – whether to immediately launch the scheduler plan or save only the definition

- **Role** - only users with the roles selected here and the administrator will be able to run the defend AI rules The selected „time frame" also affects the prediction period. If we choose "time frame = monthly", we will be able to predict a one month ahead from the moment of prediction (according to the "prediction cycle" value)

10.3 Screen content for the Trend algorithm

Algorithm:

Trend

Choose search:

Uslugi_WWW_with_cols

AI Rule Name:	rpa_trend
Feature to analyse (from search):	perf_data./
Time frame:	Day
Value type:	Average
Max probes:	10
Max predictions:	20
Data limit:	10000
Start date:	2018-03-01
Threshold:	-1
Scheduler:	☐
Role:	admin / ALL_test / audit / databases

Description of controls:

- **feature to analyze from search** - analyzed feature (dictated)

- **multiply by field** - enable multiplication of algorithms after unique values of the feature indicated here. Multiplication allows you to run the AI rule one for e.g. all servers. The value "none" in this field means no multiplication.

- **multiply by values** - if a trait is indicated in the „multiply by field", then unique values of this trait will appear in this field. Multiplications will be made for the selected values. If at least one of value is not selected, the „Run" buttons will be inactive.'

In other words, multiplication means performing an analysis for many values from the indicated field, for example: `sourece_node_host`- which we indicate in `Multiply by field (from search)`.

However, in `Multiply by values (from search)` we already indicate values of this field for which the analysis will be performed, for example: host1, host2, host3, ...

- **time frame** - feature aggregation method (1 minute, 5 minute, 15 minute, 30 minute, hourly, weekly, monthly, 6 months, 12 months)

- **max probes** - how many samples back will be taken into account for analysis. A single sample is an aggregated data according to the aggregation method.

- **value type** - which values to take into account when aggregating for a given time frame (e.g. maximum from time frame, minimum, average)

- **max predictions** - how many estimates we make for ahead (we take time frame)

- **data limit** - limits the amount of date downloaded from the source. It speeds up processing but reduces its quality

- **start date** - you can set a date earlier than the current date in order to verify how the selected algorithm would work on historical data

- **Scheduler** - a tag if the rule should be run according to the plan for the scheduler. If selected, additional fields will appear;

- **Prediction cycle** - plan definition for the scheduler, i.e. the cycle in which the prediction rule is run (e.g. once a day, every hour, once a week). In the field, enter the command that complies with the cron standard. Enable – whether to immediately launch the scheduler plan or save only the definition

- **Role** - only users with the roles selected here and the administrator will be able to run the defend AI rules The selected „time frame" also affects the prediction period. If we choose "time frame = monthly", we will be able to predict a one month ahead from the moment of prediction (according to the "prediction cycle" value)

- **Threshold** - default values -1 (do not search). Specifies the algorithm what level of exceeding the value of the feature „feature to analyze from cheese" is to look for. The parameter currently used only by the "Trend" algorithm.

10.4 Screen content for the neural network (MLP) algorithm

Descriptions of controls:

- **Name** - name of the learned neural network

- **Choose search** - search defined in ITRS Log Analytics, which is used to select a set of data on which the rule of artificial intelligence will work

- **Below**, on the left, a list of attributes and their weights based on teaching ANN will be defined during the teaching. The user for each attribute will be able to indicate the field from the above mentioned search, which contain the values of the attribute and which will be analyzed in the algorithm. The presented list (for input and output attributes) will have a static and dynamic part. Static creation by presenting key with the highest weights. The key will be presented in the original form, i.e. perf_data./ The second part is a DropDown type list that

will serve as a key update according to the user's naming. On the right side, the attribute will be examined in a given rule / pattern. Here also the user must indicate a specific field from the search. In both cases, the input and output are narrowed based on the search fields indicated in Choose search.

- **Data limit** - limits the amount of data downloaded from the source. It speeds up the processing, but reduces its quality.

- **Scheduler** - a tag if the rule should be run according to the plan or the scheduler. If selected, additional fields will appear:

- **Prediction cycle** - plan definition for the scheduler, i.e. the cycle in which the prediction rule is run (e.g. once a day, every hour, once a week). In the field, enter the command that complies with the *cron* standard

- **Enable** - whether to immediately launch the scheduler plan or save only the definition

- **Role** - only users with the roles selected here and the administrator will be able to run the defined AI rules

10.5 AI Rules List

Logged in as : logserver

Create AI Rule | **AI Rules List** | AI Learn | AI Learn Tasks

≣ AI Rules List ⟳

	Name	Search	Method	Actions
✔	int1	Uslugi_WWW_with_cols	Trend	Show Delete Update Preview
✖	k1	Uslugi_WWW_with_cols	Trend	Show Delete Update
✔	k2	Uslugi_WWW_with_cols	Trend	Show Delete Update Preview
✔	k3	Uslugi_WWW_with_cols	Trend	Show Delete Update Preview
✔	ko4	Uslugi_WWW_with_cols	Random Forest Regression Shift	Show Delete Update Preview
✔	ko5	Uslugi_WWW_with_cols	Trend	Show Delete Update Preview
✔	rpa_lrs_day_2	Linux_host_load	Linear Regression Shift Trend	Show Delete Update Preview
✔	rpa_lrst_day_100	Linux_host_load	Linear Regression Shift Trend	Show Delete Update Preview

Choose search:

Linux_host_load

Feature to analyse (from search):	perf_data.load5
Time frame:	Day
Value type:	Average
Max probes:	20
Max predictions:	10
Scheduler:	☐
Role:	admin ALL_test audit databases

⊙ (7582)	rpa_machine_state_2	Linux_host_load	Simple Moving Average	Show Delete Update
	test_sched	Linux_host_load	Simple Moving Average	Show Enable Delete Update

Column description:

- **Status**:

    - ⊙ - the process is being processed (the pid of the process is in brackets)

    - ✔ - process completed correctly

    - ✖ - the process ended with an error

- **Name** - the name of the rule

- **Search** - the search on which the rule was run

- **Method** - an algorithm used in the AI rule

- **Actions** - allowed actions:

    - **Show** - preview of the rule definition

    - **Enable/Disable** - rule activation /deactivation

    - **Delete** - deleting the rule

    - **Update** - update of the rule definition

    - **Preview** - preview of the prediction results (the action is available after the processing has been completed correctly).

10.6 AI Learn

Description of controls:

- **Search** - a source of data for teaching the network

- **prefix name** - a prefix added to the id of the learned model that allows the user to recognize the model

- **Input cols** - list of fields that are analyzed / input features. Here, the column that will be selected in the output col should not be indicated. Only those columns that are related to processing should be selected. **

- **Output col** - result field, the recognition of which is learned by the network. **This field should exist in the learning and testing data, but in the production data is unnecessary and should not occur. This field cannot be on the list of selected fields in "input col".**

- **Output class category** - here you can enter a condition in SQL format to limit the number of output categories e.g. `if((outputCol) \< 10,(floor((outputCol))+1), Double(10))`. This condition limits

the number of output categories to 10. **Such conditions are necessary for fields selected in "output col" that have continuous values. They must necessarily by divided into categories. In the Condition, use your own outputCol name instead of the field name from the index that points to the value of the "output col" attribute.**

- **Time frame** - a method of aggregation of features to improve their quality (e.g. 1 minute, 5 minutes, 15 minutes, 30 minutes, 1 hour, 1 daily).

- **Time frames output shift** - indicates how many time frame units to move the output category. This allows teaching the network with current attributes, but for categories for the future.

- **Value type** - which values to take into account when aggregating for a given time frame (e.g. maximum from time frame, minimum, average)

- **Output class count**- the expected number of result classes. **If during learning the network identifies more classes than the user entered, the process will be interrupted with an error, therefore it is better to set up more classes than less, but you have to keep in mind that this number affects the learning time.**

- **Neurons in first hidden layer (from, to)** - the number of neurons in the first hidden layer. Must have a value > 0. Jump every 1.

- **Neurons in second hidden layer (from, to)** - the number of neurons in second hidden layer. If = 0, then this layer is missing. Jump every 1.

- **Neurons in third hidden layer (from, to)** - the number of neurons in third hidden layer. If = 0 then this layer is missing. Jump every 1.

- **Max iter** (from, to) - maximum number of network teaching repetitions (the same data is used for learning many times in internal processes of the neural network). The slower it is. Jump every 100. The maximum value is 10, the default is 1.

- **Split data to train&test** - for example, the entered value of 0.8 means that the input data for the network will be divided in the ratio 0.8 to learning, 0.2 for the tests of the network learned.

- **Data limit** - limits the amount of data downloaded from the source. It speeds up the processing, but reduces its quality.

- **Max probes** - limits the number of samples taken to learn the network. Samples are already aggregated according to the selected "Time frame" parameter. It speed up teaching but reduces its quality.

- **Build** - a button to start teaching the network. The button contains the number of required teaching curses. You should be careful and avoid one-time learning for more than 1000 courses. It is better to divide them into several smaller ones. One pass after a full data load take about 1-3 minutes on a 4 core 2.4.GHz server. **The module has implemented the best practices related to the number of neurons in individual hidden layers. The values suggested by the system are optimal from the point of view of these practices, but the user can decide on these values himself.**

Under the parameters for learning the network there is an area in which teaching results will appear.

After pressing the "Refresh" button, the list of the resulting models will be refreshed.

Autorefresh - selecting the field automatically refreshes the list of learning results every 10s.

The following information will be available in the table on the left:

- **Internal name** - the model name given by the system, including the user - specified prefix

- **Overall efficiency** - the network adjustment indicator - allow to see at a glance whether it is worth dealing with the model. The grater the value, the better.

After clicking on the table row, detailed data collected during the learning of the given model will be displayed. This data will be visible in the box on the right.

The selected model can be saved under its own name using the "Save algorithm" button. This saved algorithm will be available in the "Choose AI Rule" list when creating the rule (see Create AI Rule).

10.7 AI Learn Tasks

The "AI Learn Task" tab shows the list of processes initiated teaching the ANN network with the possibility of managing processes.

Each user can see only the process they run. The user in the role of Intelligence sees all running processes.

Description of controls:

- **Algorithm prefix** - this is the value set by the user on the AI Learn screen in the Prefix name field

- **Progress** - here is the number of algorithms generated / the number of all to be generated

- **Processing time** - duration of algorithm generation in seconds (or maybe minutes or hours)

- **Actions**:

    - **Cancel** - deletes the algorithm generation task (user require confirmation of operation)

    - **Pause / Release** - pause / resume algorithm generation process.

AI Learn tab contain the Show in the preview mode of the ANN hyperparameters After completing the learning activity or after the user has interrupted it, the "Delete" button appears in "Action" field. This button allows you to permanently delete the learning results of a specific network.

10.8 Scenarios of using algorithms implemented in the Intelligence module

10.8.1 Teaching MLP networks and choosing the algorithm to use:

1. Go to the AI Learn tab,

2. We introduce the network teaching parameters,

3. Enter your own prefix for the names of the algorithms you have learned,

4. Press Build.

5. We observe the learned networks on the list (we can also stop the observation at any moment and go to other functions of the system. We will return to the learning results by going to the AI Learn Tasks tab and clicking the show action),

6. We choose the best model from our point of view and save it under our own name,

7. From this moment the algorithm is visible in the Create AI Rule tab.

10.8.2 Starting the MLP network algorithm:

1. Go to the Create AI Rule tab and create rules,

2. Select the previously saved model of the learned network,

3. Specify parameters visible on the screen (specific to MLP),

4. Press the Run button.

10.8.3 Starting regression algorithm:

1. Go to the Create AI Rule tab and create rules,

2. We choose AI Rule, e.g. Simple Moving Average, Linear Regression or Random Forest Regression, etc.,

3. Enter your own rule name (specific to regression),

4. Set the parameters of the rule (specific to regression),

5. Press the Run button.

10.8.4 Management of available rules:

1. Go to the AI Rules List tab,

2. A list of AI rules available for our role is displayed,

3. We can perform the actions available on the right for each rule.# Results of algorithms #

The results of the "AI algorithms" are saved to the index „intelligence" specially created for this purpose. The index with the prediction result. These following fields are available in the index (where xxx is the name of the attribute being analyzed):

- **xxx_pre** - estimate value

- **xxx_cur** - current value at the moment of estimation

- **method_name** - name of the algorithm used

- **rmse** - avarage square error for the analysis in which _cur values were available. **The smaller the value, the better.**

- **rmse_normalized** - mean square error for the analysis in which _cur values were available, normalized with _pre values. **The smaller the value, the better.**

- **overall_efficiency** - efficiency of the model. **The greater the value, the better. A value less than 0 may indicate too little data to correctly calculate the indicator**

- **linear_function_a** - directional coefficient of the linear function y = ax + b. **Only for the Trend and Linear Regression Trend algorithm**

- **linear_function_b** - the intersection of the line with the Y axis for the linear function y = ax + b. **Only for the Trend and Linear Regression Trend algorithm.**

Visualization and signals related to the results of data analysis should be created from this index. The index should be available to users of the Intelligence module.

10.9 Scheduler Module

ITRS Log Analytics has a built-in task schedule. In this module, we can define a command or a list of commands whose execution we instruct the application in the form of tasks. We can determine the time and frequency of tasks. Tasks can contain a simple syntax, but they can also be associated with modules, e.g. with Intelligence module.

To go to the Scheduler window, select the tile icon from the main menu bar and then go to the „Scheduler" icon (To go back, go to the „Search" icon)

Discover

Visualize

Dashboard

XLSX Import

Cluster

Wazuh

LogTrail

Agents

Alerts

Intelligence

Scheduler

Reports

Config

Dev Tools

Management
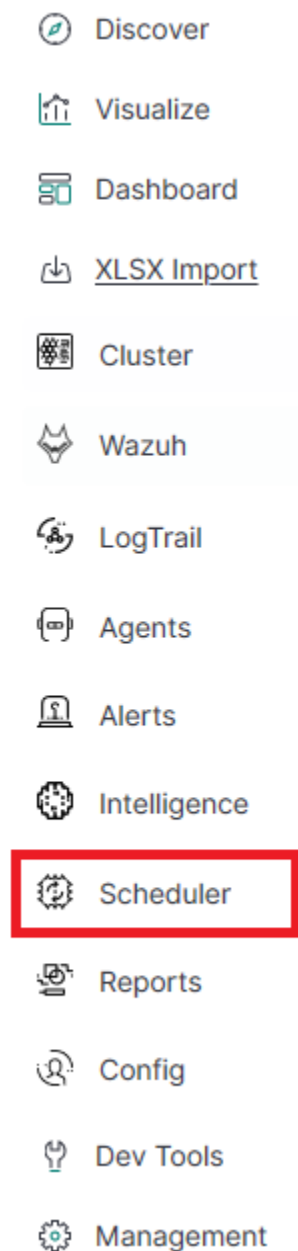
The page with three tabs will be displayed: Creating new tasks in the „Create Scheduler Job", managing tasks in the „Job List" and checking the status of tasks in „Jobs Status"

In the window for creating new tasks we have a form consisting of fields:

- **Name** - in which we enter the name of the task

- **Cron Pattern** - a field in which in cron notation we define the time and frequency of the task

- **Command** - we give the syntax of the command that will be executed in this task. These can be simple system commands, but also complex commands related to the Intelligence module. In the task management window, we can activate /deactivate, delete and update the task by clicking on the selected icon for a given task

Show   Disable

Update   Delete
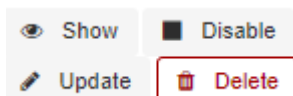
In the task status windows you can check the current status of the task: if it activated, when it started and when it ended, how long it took. This window is not editable and indicates historical data.

10.10 Permission

Permission have been implemented in the following way:

- Only the user in the admin role can create / update rules.
- When creating rules, the roles that will be able to enables / disengage / view the rules will be indicated.

We assume that the Learn process works as an administrator.

We assume that the visibility of Search in AI Learn is preceded by receiving the search permission in the module object permission.

The role of "Intelligence" launches the appropriate tabs.

An ordinary user only sees his models. The administrator sees all models.

10.11 Register new algorithm

For register new algorithm:

- **Login** to the ITRS Log Analytics
- Select **Intelligence**
- Select **Algorithm**
- Fill Create algorithm form and press **Submit** button

Form fields:

```
| Field   | Description                                                           ⌴
→                                 |
|---------|-----------------------------------------------------------------------
→-------------------------------------|
| Code    | Short name for algorithm                                              ⌴
→                                 |
| Name    | Algorithm name                                                        ⌴
→                                 |
| Command | Command to execute. The command must be in the directory pointed to by⌴
→the parameter elastscheduler.commandpath. |
```

ITRS Log Analytics execute command:

```
<command> <config> <error file> <out file>
```

Where:

- command - Command from command filed of Create algorithm form.
- config - Full path of json config file. The name of file is id of process status document in index .intelligence_rules
- error file - Unique name for error file. Not used by predefined algorithms.
- out file - Unique name for output file. Not used by predefined algorithms.

Config file:

Json document:

```
| Field                   | Value                                                    ␣
↪                         | Screen field (description)                               ␣
↪              |
|-----------------------|----------------------------------------------------------
↪-----------------------|----------------------------------------------------------
↪-----------|
| algorithm_type          | GMA, GMAL, LRS, LRST, RFRS, SMAL, SMA, TL                ␣
↪                         | Algorithm. For customs method field Code from Create␣
↪algorithm form. |
| model_name              | Not empty string.                                        ␣
↪                         | AI Rule Name.                                            ␣
↪              |
| search                  | Search id.                                               ␣
↪                         | Choose search.                                           ␣
↪              |
| label_field.field       |                                                          ␣
↪                         | Feature to analyse.                                      ␣
↪              |
| max_probes              | Integer value                                            ␣
↪                         | Max probes                                               ␣
↪              |
| time_frame              | 1 minute, 5 minutes, 15 minutes, 30 minutes, 1 hour, 1 day,
↪ 1 week, 30 day, 365 day | Time frame                                               ␣
↪              |
| value_type              | min, max, avg, count                                     ␣
↪                         | Value type                                               ␣
↪              |
| max_predictions         | Integer value                                            ␣
↪                         | Max predictions                                          ␣
↪              |
| threshold               | Integer value                                            ␣
↪                         | Threshold                                                ␣
↪              |
| automatic_cron          | Cron format string                                       ␣
↪                         | Automatic cycle                                          ␣
↪              |
| automatic_enable        | true/false                                               ␣
↪                         | Enable                                                   ␣
↪              |
| automatic               | true/false                                               ␣
↪                         | Automatic                                                ␣
↪              |
| start_date              | YYYY-MM-DD HH:mm or now                                   ␣
↪                         | Start date                                               ␣
↪              |
| multiply_by_values      | Array of string values                                   ␣
↪                         | Multiply by values                                       ␣
↪              |
| multiply_by_field       | None or full field name eg.: system.cpu                  ␣
↪                         | Multiply by field                                        ␣
↪              |
| selectedroles           | Array of roles name                                      ␣
↪                         | Role                                                     ␣
↪              |
```

(continues on next page)

```
| last_execute_timestamp |                                               ␣
→                        | Last execute                                  ␣
→               |

| Not screen fields      |                                               |
|-----------------------|-----------------------------------------------|
| preparation_date      | Document preparation date.                    |
| machine_state_uid     | AI rule machine state uid.                    |
| path_to_logs          | Path to ai machine logs.                      |
| path_to_machine_state | Path to ai machine state files.               |
| searchSourceJSON      | Query string.                                 |
| processing_time       | Process operation time.                       |
| last_execute_mili     | Last executed time in milliseconds.           |
| pid                   | Process pid if ai rule is running.            |
| exit_code             | Last executed process exit code.              |
```

The command must update the process status document in the system during operation. It is elastic partial document update.

```
| Process status         | Field (POST body)         | Description         ␣
→               |
|-----------------------|---------------------------|---------------------
→-----------------|
| START                 | doc.pid                   | System process id   ␣
→               |
|                       | doc.last_execute_timestamp | Current timestamp. yyyy-MM-dd␣
→HH:mm       |
|                       | doc.last_execute_mili     | Current timestamp in␣
→millisecunds.      |
| END PROCESS WITH ERROR | doc.error_description     | Error description.  ␣
→               |
|                       | doc.error_message         | Error message.      ␣
→               |
|                       | doc.exit_code             | System process exit code.   ␣
→               |
|                       | doc.pid                   | Value 0.            ␣
→               |
|                       | doc.processing_time       | Time of execute process in␣
→seconds.          |
| END PROCESS OK        | doc.pid                   | Value 0.            ␣
→               |
|                       | doc.exit_code             | System process exit code.␣
→Value 0 for success. |
|                       | doc.processing_time       | Time of execute process in␣
→seconds.         |
```

The command must insert data for prediction chart.

```
| Field             | Value              | Description                       ␣
→             |
|-------------------|--------------------|-----------------------------------
→-----------|
| model_name        | Not empty string.  | AI Rule Name.                     ␣
→             |
| preparationUID    | Not empty string.  | Unique prediction id              ␣
→             |
```

```
| machine_state_uid | Not empty string. | AI rule machine state uid.                  ␣
↪          |
| model_uid         | Not empty string. | Model uid from config file                  ␣
↪          |
| method_name       | Not empty string. | User friendly algorithm name.               ␣
↪          |
| <field>           | Json              | Field calculated. For example: system.cpu.
↪idle.pct_pre |
```

Document sample:

```
                {
                  "_index": "intelligence",
                  "_type": "doc",
                  "_id": "emca_TL_20190304_080802_20190531193000",
                  "_version": 2,
                  "_score": null,
                  "_source": {
                    "machine_state_uid": "emca_TL_20190304_080802",
                    "overall_efficiency": 0,
                    "processing_time": 0,
                    "rmse_normalized": 0,
                    "predictionUID": "emca_TL_20190304_080802_20190531193000",
                    "linear_function_b": 0,
                    "@timestamp": "2019-05-31T19:30:00.000+0200",
                    "linear_function_a": 0.006787878787878788,
                    "system": {
                      "cpu": {
                        "idle": {
                          "pct_pre": 0.8213333333333334
                        }
                      }
                    },
                    "model_name": "emca",
                    "method_name": "Trend",
                    "model_uid": "emca_TL_20190304_080802",
                    "rmse": 0,
                    "start_date": "2019-03-04T19:30:01.279+0100"
                  },
                  "fields": {
                    "@timestamp": [
                      "2019-05-31T17:30:00.000Z"
                    ]
                  },
                  "sort": [
                    1559323800000
                  ]
                }
```

Verification steps and logs

11.1 Verification of Elasticsearch service

To verify of Elasticsearch service you can use following command:

- Control of the Elastisearch system service via **systemd**:

```
# sysetmctl status elasticsearch
```

output:

```
    elasticsearch.service - Elasticsearch
    Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; disabled;
→vendor preset: disabled)
    Active: active (running) since Mon 2018-09-10 13:11:40 CEST; 22h ago
  Docs: http://www.elastic.co
  Main PID: 1829 (java)
    CGroup: /system.slice/elasticsearch.service
    └─1829 /bin/java -Xms4g -Xmx4g -XX:+UseConcMarkSweepGC -
→XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMSInitiatingOccupancyOnly -
→XX:+AlwaysPreTouch -Xss1m ...
```

- Control of Elasticsearch instance via **tcp port**:

```
# curl -XGET '127.0.0.1:9200/'
```

output:

```
        {
          "name" : "dY3RuYs",
          "cluster_name" : "elasticsearch",
          "cluster_uuid" : "EHZGAnJkStqlgRImqwzYQQ",
          "version" : {
            "number" : "6.2.3",
```

(continues on next page)

```
              "build_hash" : "c59ff00",
              "build_date" : "2018-03-13T10:06:29.741383Z",
              "build_snapshot" : false,
              "lucene_version" : "7.2.1",
              "minimum_wire_compatibility_version" : "5.6.0",
              "minimum_index_compatibility_version" : "5.0.0"
          },
          "tagline" : "You Know, for Search"
      }
```

- Control of Elasticsearch instance via **log file**:

```
# tail -f /var/log/elasticsearch/elasticsearch.log
```

- other control commands via *curl* **application**:

```
curl -xGET "http://localhost:9200/_cat/health?v"
curl -XGET "http://localhost:9200/_cat/nodes?v"
curl -XGET "http://localhost:9200/_cat/indicates"
```

11.2 Verification of Logstash service

To verify of Logstash service you can use following command:

- control Logstash service via **systemd**:

```
# systemctl status logstash
```

output:

```
      logstash.service - logstash
        Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor␣
→preset: disabled)
        Active: active (running) since Wed 2017-07-12 10:30:55 CEST; 1 months 23␣
→days ago
      Main PID: 87818 (java)
        CGroup: /system.slice/logstash.service
              └─87818 /usr/bin/java -XX:+UseParNewGC -XX:+UseConcMarkSweepGC
```

- control Logstash service via **port tcp**:

```
# curl -XGET '127.0.0.1:9600'
```

output:

```
      {
        "host": "skywalker",
        "version": "4.5.3",
        "http_address": "127.0.0.1:9600"
      }
```

- control Logstash service via **log file**:

```
# tail -f /var/log/logstash/logstash-plain.log
```

11.2.1 Debuging

- dynamically update logging levels through the logging API (service restart not needed):

```
curl -XPUT 'localhost:9600/_node/logging?pretty' -H 'Content-Type: application/
↪json' -d'
{
    "logger.logstash.outputs.elasticsearch" : "DEBUG"
}
'
```

- permanent change of logging level (service need to be restarted):

    - edit file */etc/logstash/logstash.yml* and set the following parameter:

    ```
    *log.level: debug*
    ```

    - restart logstash service:

    ```
            *systemctl restart logstash*
    ```

- checking correct syntax of configuration files:

```
*/usr/share/logstash/bin/logstash -tf /etc/logstash/conf.d*
```

- get information about load of the Logstash:

```
*# curl -XGET '127.0.0.1:9600/_node/jvm?pretty=true'*
```

output:

```
 {
 "host" : "logserver-test",
 "version" : "5.6.2",
 "http_address" : "0.0.0.0:9600",
 "id" : "5a440edc-1298-4205-a524-68d0d212cd55",
 "name" : "logserver-test",
 "jvm" : {
   "pid" : 14705,
   "version" : "1.8.0_161",
   "vm_version" : "1.8.0_161",
   "vm_vendor" : "Oracle Corporation",
   "vm_name" : "Java HotSpot(TM) 64-Bit Server VM",
   "start_time_in_millis" : 1536146549243,
   "mem" : {
     "heap_init_in_bytes" : 268435456,
     "heap_max_in_bytes" : 1056309248,
     "non_heap_init_in_bytes" : 2555904,
     "non_heap_max_in_bytes" : 0
   },
   "gc_collectors" : [ "ParNew", "ConcurrentMarkSweep" ]
 }                                              # Verificatoin of ITRS Log␣
↪Analytics GUI service #
```

To verify of ITRS Log Analytics GUI service you can use following command:

- control the ITRS Log Analytics GUI service via **systemd**:

```
# systemctl status kibana
```

output:

```
       kibana.service – Kibana
         Loaded: loaded (/etc/systemd/system/kibana.service; disabled; vendor␣
→preset: disabled)
         Active: active (running) since Mon 2018-09-10 13:13:19 CEST; 23h ago
       Main PID: 1330 (node)
         CGroup: /system.slice/kibana.service
                 └─1330 /usr/share/kibana/bin/../node/bin/node --no-warnings /usr/
→share/kibana/bin/../src/cli -c /etc/kibana/kibana.yml
```

- control the ITRS Log Analytics GUI via **port tcp/http**:

```
# curl -XGET '127.0.0.1:5601/'
```

output:

```
       <script>var hashRoute = '/app/kibana';
       var defaultRoute = '/app/kibana';
       var hash = window.location.hash;
       if (hash.length) {
         window.location = hashRoute + hash;
       } else {
         window.location = defaultRoute;
       }</script>
```

- Control the ITRS Log Analytics GUI via **log file**:

```
# tail -f /var/log/messages
```

Building a cluster

12.1 Node roles

Every instance of Elasticsearch server is called a *node*. A collection of connected nodes is called a *cluster*. All nodes know about all the other nodes in the cluster and can forward client requests to the appropriate node.

Besides that, each node serves one or more purpose:

- Master-eligible node - A node that has *node.master* set to true (default), which makes it eligible to be elected as the master node, which controls the cluster

- Data node - A node that has *node.data* set to true (default). Data nodes hold data and perform data related operations such as CRUD, search, and aggregations

- Client node - A client node has both *node.master* and *node.data* set to false. It can neither hold data nor become the master node. It behaves as a "*smart router*" and is used to forward cluster-level requests to the master node and data-related requests (such as search) to the appropriate data nodes

- Tribe node - A tribe node, configured via the *tribe.** settings, is a special type of client node that can connect to multiple clusters and perform search and other operations across all connected clusters.

12.2 Naming convention

Elasticsearch require little configuration before before going into work.

The following settings must be considered before going to production:

- **path.data** and **path.logs** - default locations of these files are: `/var/lib/elasticsearch` and `/var/log/elasticsearch`.

- **cluster.name** - A node can only join a cluster when it shares its `cluster.name` with all the other nodes in the cluster. The default name is "elasticsearch", but you should change it to an appropriate name which describes the purpose of the cluster. You can do this in `/etc/elasticsearch/elasticsearch.yml` file.

- **node.name** - By default, Elasticsearch will use the first seven characters of the randomly generated UUID as the node id. Node id is persisted and does not change when a node restarts. It is worth configuring a more human readable name: `node.name:  prod-data-2` in file `/etc/elstaicsearch/elasticsearch.yml`

- **network.host** - parametr specifying network interfaces to which Elasticsearch can bind. Default is `network.host:  ["_local_","_site_"].`

- **discovery** - Elasticsearch uses a custom discovery implementation called "Zen Discovery". There are two important settings:

  - `discovery.zen.ping.unicast.hosts` - specify list of other nodes in the cluster that are likely to be live and contactable;

  - `discovery.zen.minimum_master_nodes` - to prevent data loss, you can configure this setting so that each master-eligible node knows the minimum number of master-eligible nodes that must be visible in order to form a cluster.

- **heap size** - By default, Elasticsearch tells the JVM to use a heap with a minimum (Xms) and maximum (Xmx) size of 1 GB. When moving to production, it is important to configure heap size to ensure that Elasticsearch has enough heap available

12.3 Config files

To configure the Elasticsearch cluster you must specify some parameters in the following configuration files on every node that will be connected to the cluster:

- `/etc/elsticsearch/elasticserach.yml:`

  - `cluster.name:name_of_the_cluster` - same for every node;

  - `node.name:name_of_the_node` - uniq for every node;

  - `node.master:true_or_false`

  - `node.data:true_or_false`

  - `network.host:["_local_","_site_"]`

  - `discovery.zen.ping.multicast.enabled`

  - `discovery.zen.ping.unicast.hosts`

- `/etc/elsticsearch/log4j2.properties:`

  - `logger:  action:  DEBUG` - for easier debugging.

12.4 Example setup

Example of the Elasticsearch cluster configuration:

- file `/etc/elasticsearch/elasticsearch.yml:`

```
cluster.name: tm-lab
node.name: "elk01"
node.master: true
node.data: true
network.host: 127.0.0.1,10.0.0.4
http.port: 9200
```

(continues on next page)

```
discovery.zen.ping.multicast.enabled: false
discovery.zen.ping.unicast.hosts: ["10.0.0.4:9300","10.0.0.5:9300","10.0.0.
↪6:9300"]
```

- to start the Elasticsearch cluster execute command:

```
# systemctl restart elasticsearch
```

- to check status of the Elstaicsearch cluster execute command:

    - check of the Elasticsearch cluster nodes status via tcp port:

```
# curl -XGET '127.0.0.1:9200/_cat/nodes?v'

host                    ip              heap.percent ram.percent load node.role␣
↪master name
10.0.0.4        10.0.0.4        18          91                  0.00 -    ␣
↪     -      elk01
10.0.0.5        10.0.0.5        66          91                  0.00 d    ␣
↪     *      elk02
10.0.0.6        10.0.0.6        43          86                  0.65 d    ␣
↪     m      elk03
10.0.0.7        10.0.0.7        45          77                  0.26 d    ␣
↪     m      elk04
```

    - check status of the Elasticsearch cluster via log file:

```
# tail -f /var/log/elasticsearch/tm-lab.log (cluster.name)
```

12.5 Adding a new node to existing cluster

Install the new ITRS Log Analytics instance. The description of the installation can be found in the chapter "First configuration steps"

Change the following parameters in the configuration file:

- `cluster.name`:name_of_the_cluster same for every node;

- `node.name`:name_of_the_node uniq for every node;

- `node.master`:true_or_false

- `node.data`:true_or_false

- `discovery.zen.ping.unicast.hosts`:["10.0.0.4:9300","10.0.0.5:9300","10.0.0.6:9300"] - IP addresses and instances of nodes in the cluster.

If you add a node with the role `data`, delete the contents of the `path.data` directory, by default in `/var/lib/elasticsearch`

Restart the Elasticsearch instance of the new node:

```
systemctl restart elasticsearch
```

Integration with AD

You can configure the ITRS Log Analytics to communicate with Active Directory to authenticate users. To integrate with Active Directory, you configure an Active Directory realm and assign Active Directory users and groups to the ITRS Log Analytics roles in the role mapping file.

To protect passwords, communications between the ITRS Log Analytics and the LDAP server should be encrypted using SSL/TLS. Clients and nodes that connect via SSL/TLS to the LDAP server need to have the LDAP server's certificate or the server's root CA certificate installed in their keystore or truststore.

13.1 AD configuration

The AD configuration should be done in the `/etc/elasticsearch/properties.yml` file.

Below is a list of settings to be made in the `properties.yml` file (the commented section in the file in order for the AD settings to start working, this fragment should be uncommented):

```
|**Direcitve**                                          | **Description**              ␣
↪                                                        |
| ----------------------------------------------------|----------------------------
↪--------------------------------------------------------|
| # LDAP                                                |                             ␣
↪                                                        |
| #ldaps:                                               |                             ␣
↪                                                        |
| # - name: \"example.com\"                             |# domain that is configured ␣
↪                                                        |
| # host: \"127.0.0.1,127.0.0.2\"                       |# list of server for this␣
↪domain                                                  |
| # port: 389                                           |# optional, default 389 for␣
↪unencrypted session or 636 for encrypted sessions       |
|# ssl\_enabled: false                                  |# optional, default true     ␣
↪                                                        |
|# ssl\_trust\_all\_certs: true                         |# optional, default false    ␣
↪                                                        |
```

```
|# ssl.keystore.file: \"path\"                         |# path to the truststore␣
↪store                                                 |
|# ssl.keystore.password: \"path\"                     |# password to the trusted␣
↪certificate store                                     |
|# bind\_dn: [[admin\@example.com]                     |# account name administrator␣
↪                                                      |
|# bind\_password: \"password\"                        |# password for the␣
↪administrator account                                 |
|# search\_user\_base\_DN: \"OU=lab,DC=example,DC=com\" |# search for the DN user␣
↪tree database                                         |
|# user\_id\_attribute: \"uid                          |# search for a user␣
↪attribute optional, by default \"uid\"                |
|# search\_groups\_base\_DN:\"OU=lab,DC=example,DC=com\"|# group database search.␣
↪This is a catalog main, after which the groups will be sought.|
|# unique\_member\_attribute: \"uniqueMember\"          |# optional, default\
↪"uniqueMember\"                                       |
|# connection\_pool\_size: 10                          |# optional, default␣
↪30                                                    |
|# connection\_timeout\_in\_sec: 10                    |# optional, default␣
↪1                                                     |
|# request\_timeout\_in\_sec: 10                       |# optional, default␣
↪1                                                     |
|# cache\_ttl\_in\_sec: 60                             |# optional, default 0 -␣
↪cache disabled                                        |
```

If we want to configure multiple domains, then in this configuration file we copy the # LDAP section below and configure it for the next domain.

Below is an example of how an entry for 2 domains should look like. (It is important to take the interpreter to read these values correctly).

```
ldaps:
        - name: "example1.com"
          host: "127.0.0.1,127.0.0.2"
          port: 389 # optional, default 389
          ssl_enabled: false # optional, default true
          ssl_trust_all_certs: true # optional, default false
          bind_dn: "admin@example1.com"
          bind_password: "password" # generate encrypted password with /usr/share/
↪elasticsearch/pass-encrypter/pass-encrypter.sh
          search_user_base_DN: "OU=lab,DC=example1,DC=com"
          user_id_attribute: "uid" # optional, default "uid"
          search_groups_base_DN: "OU=lab,DC=example1,DC=com"
          unique_member_attribute: "uniqueMember" # optional, default "uniqueMember"
          connection_pool_size: 10 # optional, default 30
          connection_timeout_in_sec: 10 # optional, default 1
          request_timeout_in_sec: 10 # optional, default 1
          cache_ttl_in_sec: 60 # optional, default 0 - cache disabled
          service_principal_name: "esauth@example1.com" # optional, for sso
          service_principal_name_password : "password" # optional, for sso
        - name: "example2.com" #DOMAIN 2
          host: "127.0.0.1,127.0.0.2"
          port: 389 # optional, default 389
          ssl_enabled: false # optional, default true
          ssl_trust_all_certs: true # optional, default false
          bind_dn: "admin@example2.com"
          bind_password: "password" # generate encrypted password with /usr/share/
↪elasticsearch/pass-encrypter/pass-encrypter.sh
```

```
        search_user_base_DN: "OU=lab,DC=example2,DC=com"
        user_id_attribute: "uid" # optional, default "uid"
        search_groups_base_DN: "OU=lab,DC=example2,DC=com"
        unique_member_attribute: "uniqueMember" # optional, default "uniqueMember"
        connection_pool_size: 10 # optional, default 30
        connection_timeout_in_sec: 10 # optional, default 1
        request_timeout_in_sec: 10 # optional, default 1
        cache_ttl_in_sec: 60 # optional, default 0 – cache disabled
        service_principal_name: "esauth@example2.com" # optional, for sso
        service_principal_name_password : "password" # optional, for ssl
```

After completing the LDAP section entry in the `properties.yml` file, save the changes and restart the service with the command:

```
# systemctl restart elasticsearch
```

13.2 Configure SSL suport for AD authentication

Open the certificate manager on the AD server.
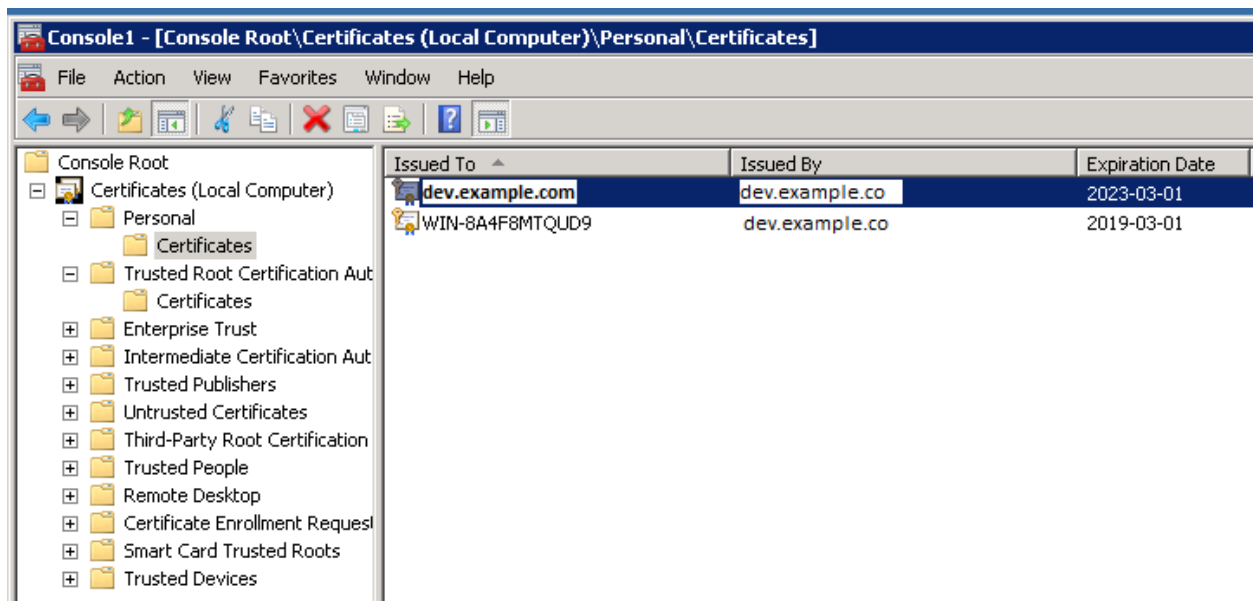
Select the certificate and open it
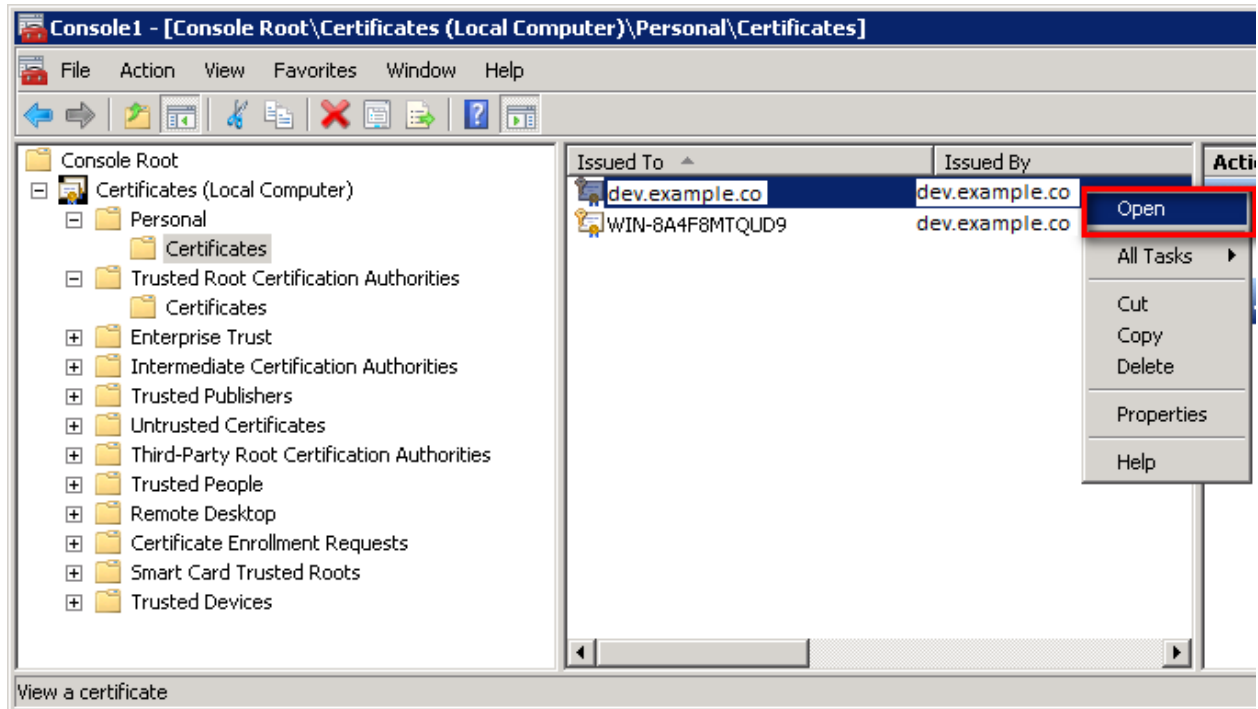
Select the option of copying to a file in the Details tab
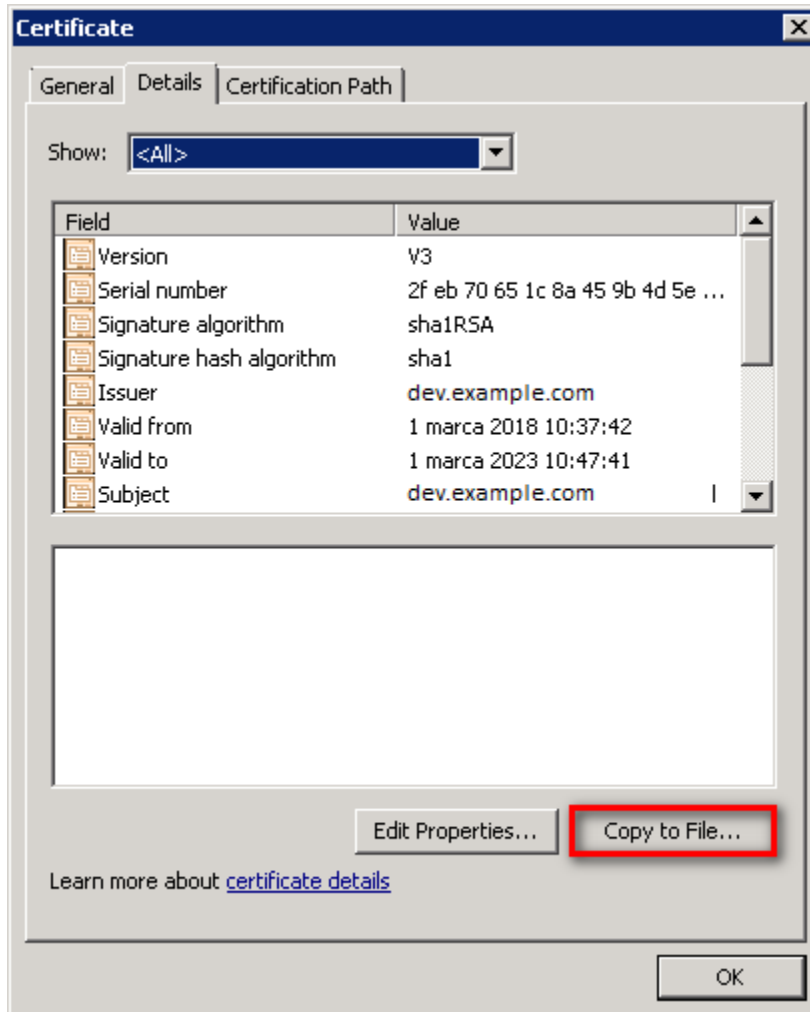
Click the Next button

Keep the setting as shown below and click Next
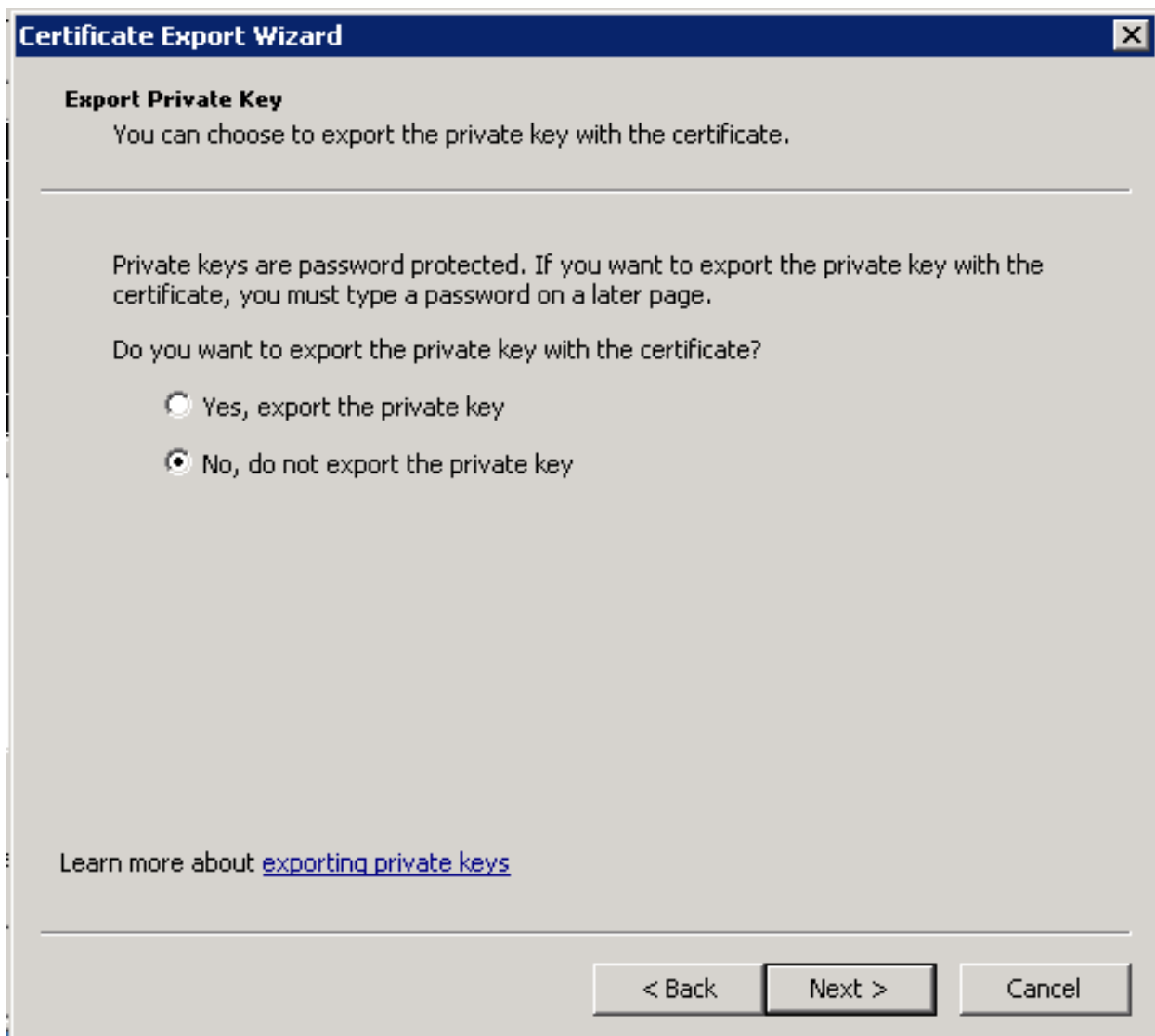
Keep the setting as shown below and click Next.
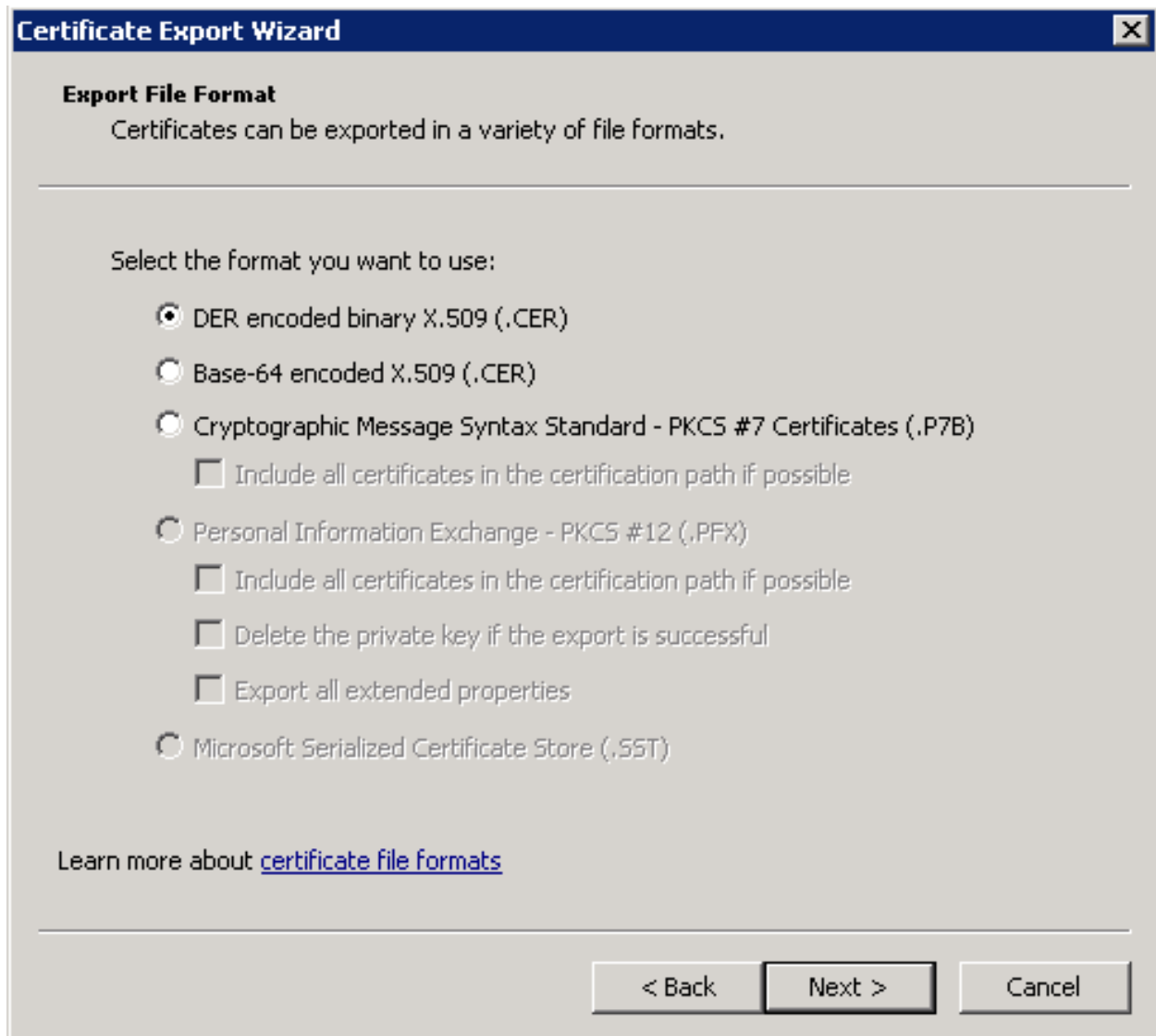
Give the name a certificate
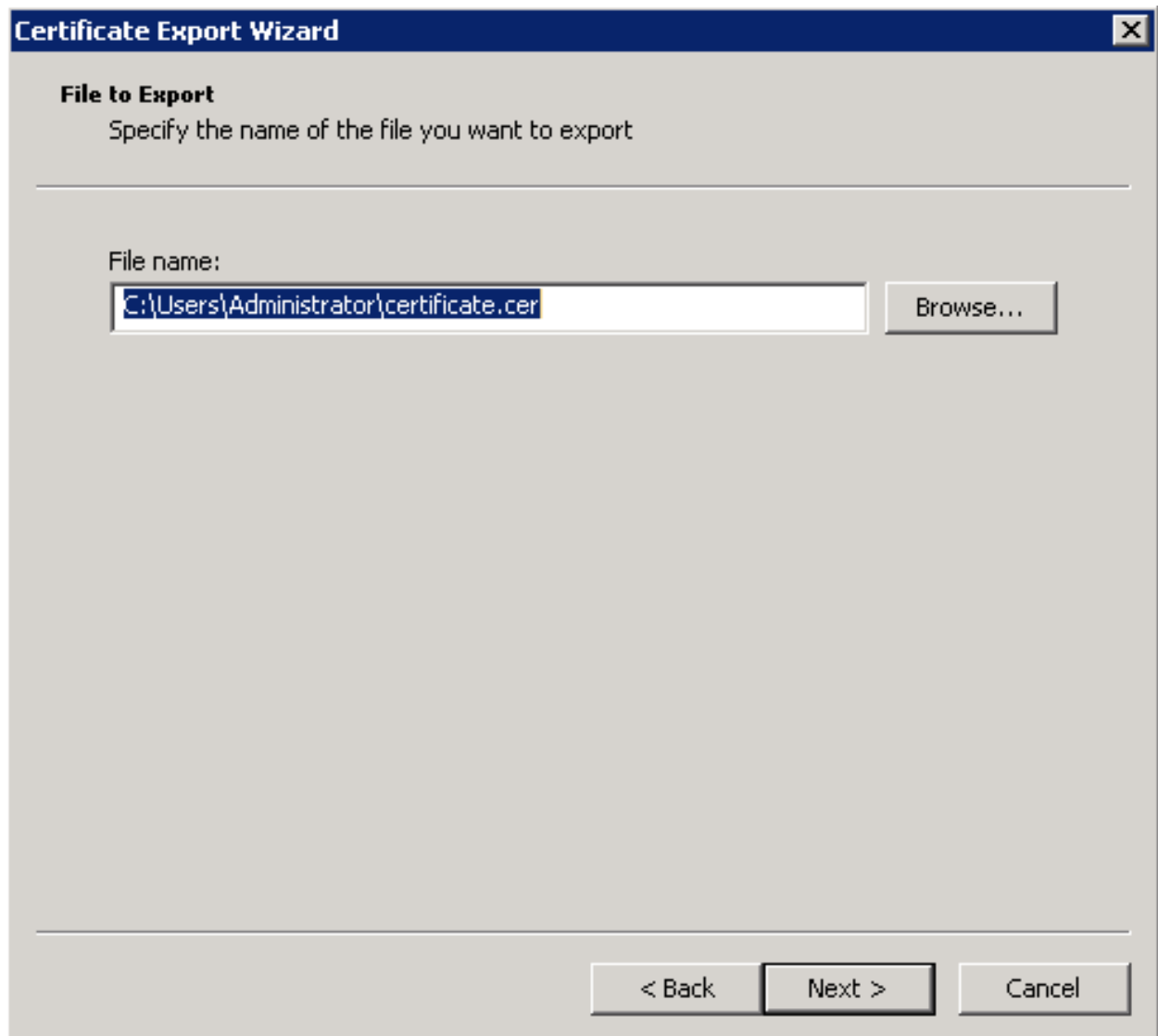
After the certificate is exported, this certificate should be imported into a trusted certificate file that will be used by the Elasticsearch plugin.

To import a certificate into a trusted certificate file, a tool called „keytool.exe" is located in the JDK installation directory.

Use the following command to import a certificate file:

```
keytool -import -alias adding_certificate_keystore -file certificate.cer -keystore
→certificatestore
```

The values for RED should be changed accordingly.

By doing this, he will ask you to set a password for the trusted certificate store. Remember this password, because it must be set in the configuration of the Elasticsearch plugin. The following settings must be set in the `properties.yml` configuration for SSL:

```
ssl.keystore.file: "<path to the trust certificate store>"
ssl.keystore.password: "< password to the trust certificate store>"
```

13.3 Role mapping

In the `/etc/elasticsearch/properties.yml` configuration file you can find a section for configuring role mapping:

```
# LDAP ROLE MAPPING FILE`
# rolemapping.file.path: /etc/elasticsearch/role-mappings.yml
```

This variable points to the file `/etc/elasticsearch/role-mappings.yml` Below is the sample content for this file:

```
admin:
"CN=Admins,OU=lab,DC=dev,DC=it,DC=example,DC=com"
bank:
"CN=security,OU=lab,DC=dev,DC=it,DC=example,DC=com"
```

Attention. The role you define in the `role.mapping` file must be created in the ITRS Log Analytics.

How to the mapping mechanism works ? An AD user log in to ITRS Log Analytics. In the application there is a admin role, which through the file role-mapping .yml binds to the name of the admin role to which the Admins container from AD is assigned. It is enough for the user from the AD account to log in to the application with the privileges that are assigned to admin role in the ITRS Log Analytics. At the same time, if it is the first login in the ITRS Log Analytics, an account is created with an entry that informs the application administrator that is was created by logging in with AD.

Similar, the mechanism will work if we have a role with an arbitrary name created in ITRS Log Analytics Logistics and connected to the name of the role-mappings.yml and existing in AD any container.

Below a screenshot of the console on which are marked accounts that were created by uesrs logging in from AD

If you map roles with from several domains, for example dev.examloe1.com, dev.example2.com then in User List we will see which user from which domain with which role logged in ITRS Log Analytics.

13.4 Password encryption

For security reason you can provide the encrypted password for Active Directory integration. To do this use *pass-encrypter.sh* script that is located in the *Utils* directory in installation folder.

1. Installation of *pass-encrypter*

```
cp -pr /instalation_folder/elasticsearch/pass-encrypter /usr/share/elasticsearch/
```

2. Use *pass-encrypter*

```
# /usr/share/elasticsearch/pass-encrypter/pass-encrypter.sh
Enter the string for encryption :
new_password
Encrypted string : MTU1MTEwMDcxMzQzMg==1GEG8KUOgyJko0PuT2C4uw==
```

Integration with Radius

To use the Radius protocol, install the latest available version of ITRS Log Analytics.

14.1 Configuration

The default configuration file is located at /etc/elasticsearch/properties.yml:

```
# Radius opts
#radius.host: "10.4.3.184"
#radius.secret: "querty1q2ww2q1"
#radius.port: 1812
```

Use appropriate secret based on config file in Radius server. The secret is configured on `clients.conf` in Radius server.

In this case, since the plugin will try to do Radius auth then client IP address should be the IP address where the Elasticsearch is deployed.

Every user by default at present get the admin role.

CHAPTER 15

Integration with LDAP

To use OpenLDAP authorization, install or update ITRS Log Analytics 7.0.2.

15.1 Configuration

The default configuration file is located at `/etc/elasticsearch/properties.yml`:

- ldap_groups_search - Enable Open LDAP authorization. The `ldap_groups_search` switch with true / false values.

- search filter - you can define `search_filter` for each domain. When polling the LDAP / AD server, the placeholder is changed to `userId` (everything before @domain) of the user who is trying to login. Sample search_filter:

```
search_filter: "(&(objectClass=inetOrgPerson)(cn=%s))"
```

If no search_filter is given, the default will be used:

```
(&(&(objectCategory=Person)(objectClass=User))(samaccountname=%s))
```

- max_connections - for each domain (must be> = 1), this is the maximum number of connections that will be created with the LDAP / AD server for a given domain. Initially, one connection is created, if necessary another, up to the maximum number of connections set. If max_connections is not given, the default value = 10 will be used.

- ldap_groups_search - filter will be used to search groups on the AD / LDAP server of which the user is trying to login. An example of groups_search_filter that works quite universally is:

```
groups_search_filter: "(|(uniqueMember=%s)(member=%s))"
```

Sample configuration:

```
licenseFilePath: /usr/share/elasticsearch/

ldaps:

    - name: "dev.it.example.com"
      host: "192.168.0.1"
      port: 389                                              # optional,␣
→default 389
      #ssl_enabled: false                                    # optional,␣
→default true
      #ssl_trust_all_certs: true                             # optional,␣
→default false
      bind_dn: "Administrator@dev2.it.example.com"
      bind_password: "Buspa#mexaj1"
      search_user_base_DN: "OU=lab,DC=dev,DC=it,DC=example,DC=pl"
      search_filter: "(&(objectClass=inetOrgperson)(cn=%s))"     # optional,␣
→default "(&(&(objectCategory=Person)(objectClass=User))(samaccountname=%s))"
      user_id_attribute: "uid"                               # optional,␣
→default "uid"
      search_groups_base_DN: "OU=lab,DC=dev,DC=it,DC=example,DC=pl" # base DN,␣
→which will be used for searching user's groups in LDAP tree
      groups_search_filter: "(member=%s)"                       # optional,␣
→default (member=%s), if ldap_groups_search is set to true, this filter will be␣
→used for searching user's membership of LDAP groups
      ldap_groups_search: false                              # optional,␣
→default false – user groups will be determined basing on user's memberOf␣
→attribute
      unique_member_attribute: "uniqueMember"                # optional,␣
→default "uniqueMember"
      max_connections: 10                                    # optional,␣
→default 10
      connection_timeout_in_sec: 10                          # optional,␣
→default 1
      request_timeout_in_sec: 10                             # optional,␣
→default 1
      cache_ttl_in_sec: 60                                   # optional,␣
→default 0 – cache disabled
```

When the password is longer than 20 characters, we recommend using our pass-encrypter, otherwise backslash must be escaped with another backslash. Endpoint `role-mapping/_reload` has been changed to `_role-mapping/reload`. This is a unification of API conventions, in accordance with Elasticsearch conventions.

Configuring Single Sign On (SSO)

In order to configure SSO, the system should be accessible by domain name URL, not IP address nor localhost.

Ok :`https://loggui.com:5601/login`. **Wrong :** `https://localhost:5601/login`, `https://10.0.10.120:5601/login`

In order to enable SSO on your system follow below steps. The configuration is made for AD: `dev.example.com`, GUI URL: `loggui.com`

16.1 Configuration steps

16.1.1 Create an User Account for Elasticsearch auth plugin

In this step, a Kerberos Principal representing Elasticsearch auth plugin is created on the Active Directory. The principal name would be `name@DEV.EXAMPLE.COM`, while the `DEV.EXAMPLE.COM` is the administrative name of the realm. In our case, the principal name will be `esauth@DEV.EXAMPLE.COM`.

Create User in AD. Set "Password never expires" and "Other encryption options" as shown below:

16.1.2 Define Service Principal Name (SPN) and Create a Keytab file for it

Use the following command to create the keytab file and SPN:

> C:\> ktpass -out c:\Users\Administrator**esauth.keytab** -princ **HTTP/loggui.com@DEV.EXAMPLE.COM**
> -mapUser **esauth** -mapOp set -pass '**Sprint$123**' -crypto ALL -pType KRB5_NT_PRINCIPAL

Values highlighted in bold should be adjusted for your system. The `esauth.keytab` file should be placed on your elasticsearch node - preferably /etc/elasticsearch/ with read permissions for elasticsearch user: `chmod 640 /etc/elasticsearch/esauth.keytab chown elasticsearch: /etc/elasticsearch/esauth.keytab`

16.1.3 Create a file named *krb5Login.conf*:

```
com.sun.security.jgss.initiate{
    com.sun.security.auth.module.Krb5LoginModule required
    principal="esauth@DEV.EXAMPLE.COM" useKeyTab=true
    keyTab=/etc/elasticsearch/esauth.keytab storeKey=true debug=true;
    };
com.sun.security.jgss.krb5.accept {
    com.sun.security.auth.module.Krb5LoginModule required
    principal="esauth@DEV.EXAMPLE.COM" useKeyTab=true
    keyTab=/etc/elasticsearch/esauth.keytab storeKey=true debug=true;
    };
```

Principal user and keyTab location should be changed as per the values created in the step 2. Make sure the domain is in UPPERCASE as shown above. The `krb5Login.conf` file should be placed on your elasticsearch node, for instance /etc/elasticsearch/ with read permissions for elasticsearch user:

```
sudo chmod 640 /etc/elasticsearch/krb5Login.conf
sudo chown elasticsearch: /etc/elasticsearch/krb5Login.conf
```

16.1.4 Append the following JVM arguments (on Elasticsearch node in */etc/sysconfig/elasticsearch*):

> -Dsun.security.krb5.debug=true -Djava.security.krb5.realm=**DEV.EXAMPLE.COM** -Djava.security.krb5.kdc=**AD_HOST_IP_ADDRESS** -Djava.security.auth.login.config=**/etc/elasticsearch/krb5Login.conf** -Djavax.security.auth.useSubjectCredsOnly=false

Change the appropriate values in the bold. This JVM arguments has to be set for Elasticsearch server.

16.1.5 Add the following additional (sso.domain, service_principal_name, service_principal_name_password) settings for ldap in elasticsearch.yml or properties.yml file wherever the ldap settings are configured:

```
sso.domain: "dev.example.com"
ldaps:
- name: "dev.example.com"
    host: "IP_address"
    port: 389                                                 # optional, default 389
    ssl_enabled: false                                        # optional, default   ␣
↪true
    ssl_trust_all_certs: false                                 # optional, default␣
↪false
    bind_dn: "Administrator@dev.example.com"                     # optional, skip for␣
↪anonymous bind
    bind_password: "administrator_password"                                 #␣
↪optional, skip for anonymous bind
    search_user_base_DN: "OU=lab,DC=dev,DC=it,DC=example,DC=com"
    user_id_attribute: "uid"                                  # optional, default "uid
↪"
    search_groups_base_DN: "OU=lab,DC=dev,DC=it,DC=example,DC=com"
    unique_member_attribute: "uniqueMember"                   # optional, default
↪"uniqueMember"
    service_principal_name: "esauth@DEV.EXAMPLE.COM"
    service_principal_name_password : "Sprint$123"
```

Note: At this moment, SSO works for only single domain. So you have to mention for what domain SSO should work in the above property `sso.domain`

16.1.6 To apply the changes restart Elasticsearch service

`sudo systemctl restart elasticsearch.service`

16.1.7 Enable SSO feature in `kibana.yml` file:

`kibana.sso_enabled: true` After that Kibana has to be restarted: `sudo systemctl restart kibana.service`

16.2 Client (Browser) Configuration##

16.2.1 Internet Explorer configuration

1. Goto `Internet Options` from `Tools` menu and click on `Security` Tab:
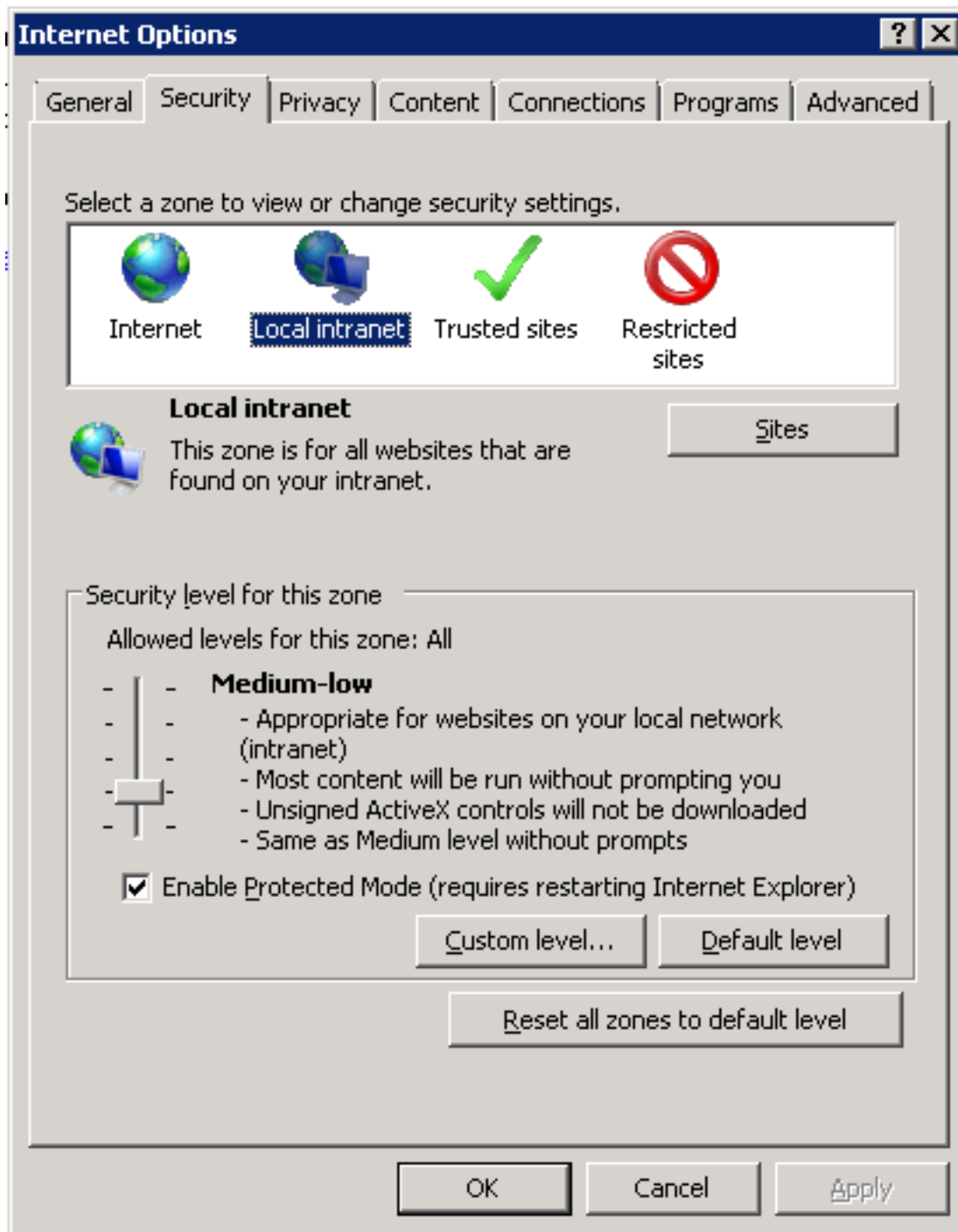
1. Select `Local intranet`, click on `Site` -> `Advanced` -> `Add` the url:
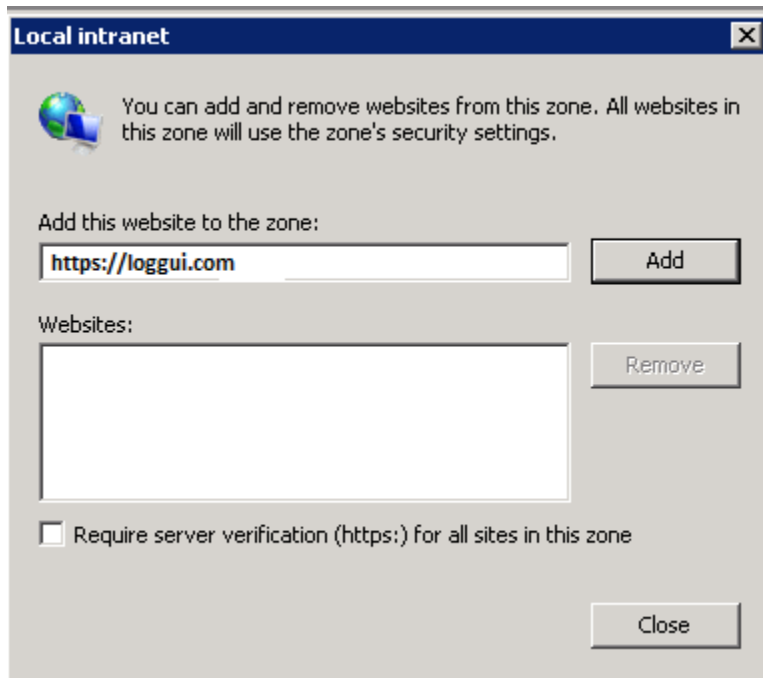
After adding the site click close.

1. Click on custom level and select the option as shown below:
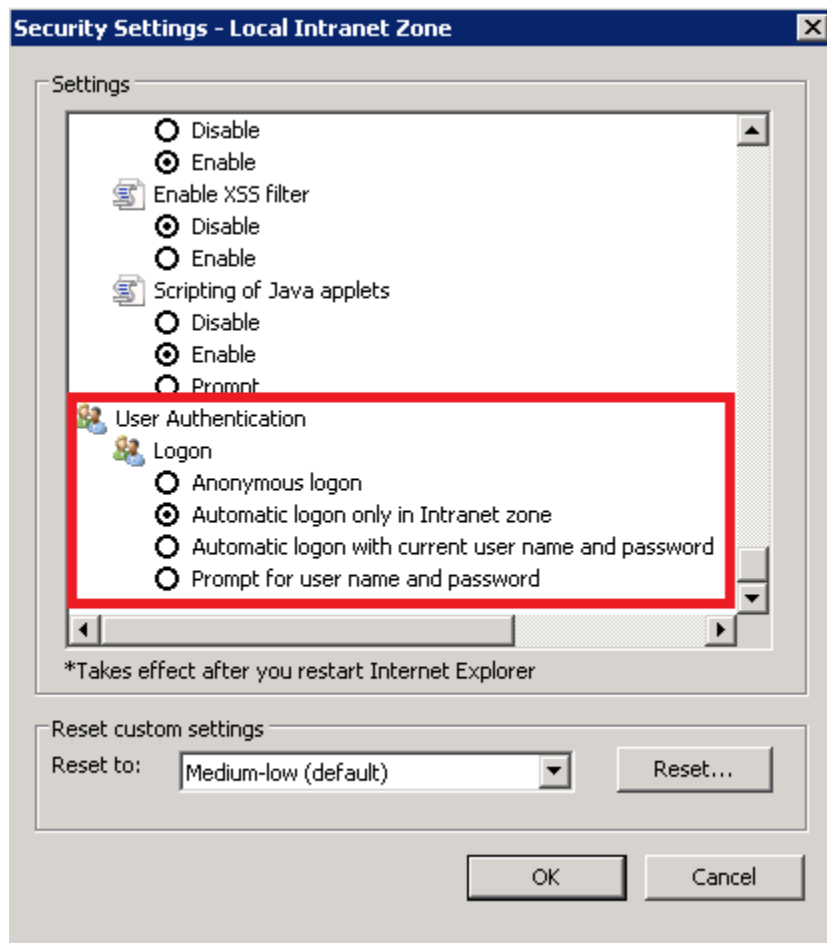
16.2.2 Chrome configuration

For Chrome, the settings are taken from IE browser.

16.2.3 Firefox configuration

Update the following config:
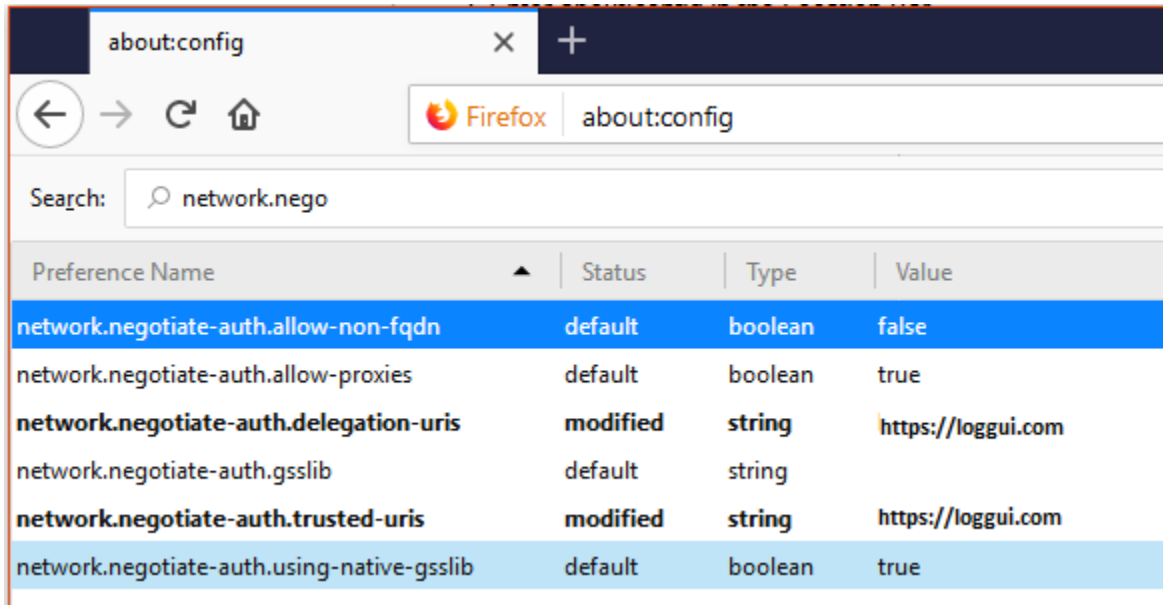
Configure email delivery

17.1 Configure email delivery for sending PDF reports in Scheduler.

The default e-mail client that installs with the Linux CentOS system, which is used by ITRS Log Analytics to send reports (Section 5.3 of the Reports chapter), is *postfix*.# Configuration file for **postfix** mail client #

The *postfix* configuration directory for CentOS is */etc/postfix*. It contains files:

main.cf - the main configuration file for the program specifying the basics parameters

Some of its directives:

```
|**Directive**         |        **Description**                                       ⎵
↪                                         |
| ----------------------|  --------------------------------------------------------
↪-----------------------------------------------|
|queue\_directory       |        The postfix queue location.
|command\_directory     |       The location of Postfix commands.
|daemon\_directory      |        Location of Postfix daemons.
|mail\_owner            |        The owner of Postfix domain name of the server
|myhostname             |        The fully qualified domain name of the server.
|mydomain               |        Server domain
|myorigin               |        Host or domain to be displayed as origin on email⎵
↪leaving the server.
|inet\_interfaces       |        Network interface to be used for incoming email.
|mydestination          |        Domains from which the server accepts mail.
|mynetworks             |        The IP address of trusted networks.
|relayhost              |        Host or other mail server through which mail will⎵
↪be sent. This server will act as an outbound gateway.
|alias\_maps            |        Database of asliases used by the local delivery⎵
↪agent.
|alias\_database        |        Alias database generated by the new aliases⎵
↪command.
|mail\_spool\_directory |        The location where user boxes will be stored.
```

master.cf - defines the configuration settings for the master daemon and the way it should work with other agents to deliver mail. For each service installed in the master.cf file there are seven columns that define how the service should be used.

```
|Column          |     Description
|-------------- | -----------------------------------------------------------
↪---------------------------
|service         |     The name of the service
|type            |     The transport mechanism to be user.
|private         |     Is the service only for user by Postfix.
|unpriv          |     Can the service be run by ordinary users
|chroot          |     Whether the service is to change the main directory (chroot)␣
↪for the mail. Queue.
|wakeup          |     Wake up interval for the service.
|maxproc         |     The maximum number of processes on which the service can be␣
↪forked (to divide in branches)
|command + args  |   A command associated with the service plus any argument
```

access - can be used to control access based on e-mail address, host address, domain or network address.

Examples of entries in the file

```
|Description                                     | Example
|------------------------------------------------|--------------------
|To allow access for specific IP address:        | 192.168.122.20 OK
|To allow access for a specific domain:          | example.com OK
|To deny access from the 192.168.3.0/24 network: | 192.168.3 REJECT
```

After making changes to the access file, you must convert its contents to the access.db database with the postmap command:

```
        # postmap /etc/postfix/access
        # ll /etc/postfix/access*

        -rw-r\--r\--. 1 root root 20876 Jan 26 2014 /etc/postfix/access
        -rw-r\--r\--. 1 root root 12288 Feb 12 07:47 /etc/postfix/access.db
```

canonical - mapping incoming e-mails to local users.

Examples of entries in the file:

To forward emails to user1 to the [[user1@yahoo.com] mailbox:

```
        user1 user1\@yahoo.com
```

To forward all emails for example.org to another example.com domain:

```
        @example.org @example.com
```

After making changes to the canonical file, you must convert its contents to the canonical.db database with the postmap command:

```
# postmap /etc/postfix/canonical
# ll /etc/postfix/canonical*

-rw-r\--r\--. 1 root root 11681 2014-06-10 /etc/postfix/canonical
-rw-r\--r\--. 1 root root 12288 07-31 20:56 /etc/postfix/canonical.db
```

generic - mapping of outgoing e-mails to local users. The syntax is the same as a canonical file. After you make change to this file, you must also run the postmap command.

```
# postmap /etc/postfix/generic
# ll /etc/postfix/generic*

-rw-r\--r\--. 1 root root 9904 2014-06-10 /etc/postfix/generic
-rw-r\--r\--. 1 root root 12288 07-31 21:15 /etc/postfix/generic.db
```

reloceted – information about users who have been transferred. The syntax of the file is the same as canonical and generic files.

Assuming tha user1 was moved from example.com to example.net, you can forward all emails received on the old address to the new address:

Example of an entry in the file:

```
user1@example.com user1@example.net
```

After you make change to this file, you must also run the postmap command.

```
# postmap /etc/postfix/relocated
# ll /etc/postfix/relocated*

-rw-r\--r\--. 1 root root 6816 2014-06-10 /etc/postfix/relocated
-rw-r\--r\--. 1 root root 12288 07-31 21:26 /etc/postfix/relocated.d
```

transport – mapping between e-mail addresses and server through which these e-mails are to be sent (next hops) int the transport format: nexthop.

Example of an entry in the file:

```
user1@example.com smtp:host1.example.com
```

After you make changes to this file, you must also run the postmap command.

```
# postmap /etc/postfix/transport
[root@server1 postfix]# ll /etc/postfix/transport*

-rw-r\--r\--. 1 root root 12549 2014-06-10 /etc/postfix/transport
-rw-r\--r\--. 1 root root 12288 07-31 21:32 /etc/postfix/transport.db
```

virtual - user to redirect e-mails intended for a certain user to the account of another user or multiple users. It can also be used to implement the domain alias mechanism.

Examples of the entry in the file:

Redirecting email for user1, to root users and user3:

```
user1 root, user3
```

Redirecting email for user 1 in the example.com domain to the root user:

```
user1@example.com root
```

After you make change to this file, you must also run the postmap command:

```
# postmap /etc/postfix/virtual
# ll /etc/postfix/virtual
```

(continues on next page)

```
-rw-r\--r\--. 1 root root 12494 2014-06-10 /etc/postfix/virtual
-rw-r\--r\--. 1 root root 12288 07-31 21:58 /etc/postfix/virtual.db
```

17.2 Basic *postfix* configuration

Base configuration of *postfix* application you can make in `/etc/postfix/main.cfg` configuration file, which must complete with the following entry:

- section *# RECEIVING MAIL*

  ```
  inet_interfaces = all
  inet_protocols = ipv4
  ```

- section *# INTERNET OR INTRANET*

  ```
  relayhost = [IP mail server]:25 (port number)
  ```

I the netxt step you must complete the canonical file of *postfix*

At the end you should restart the *postfix*:

```
systemctl restart postfix
```

17.3 Example of postfix configuration with SSL encryption enabled

To configure email delivery with SSL encryption you need to make the following changes in the *postfix* configuration files:

- **/etc/postfix/main.cf** - file should contain the following entries in addition to standard (unchecked entries):

  ```
  mydestination = $myhostname, localhost.$mydomain, localhost
  myhostname = example.com
  relayhost = [smtp.example.com]:587
  smtp_sasl_auth_enable = yes
  smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
  smtp_sasl_security_options = noanonymous
  smtp_tls_CAfile = /root/certs/cacert.cer
  smtp_use_tls = yes
  smtp_sasl_mechanism_filter = plain, login
  smtp_sasl_tls_security_options = noanonymous
  canonical_maps = hash:/etc/postfix/canonical
  smtp_generic_maps = hash:/etc/postfix/generic
  smtpd_recipient_restrictions = permit_sasl_authenticated
  ```

- **/etc/postfix/sasl/passwd** - file should define the data for authorized

  ```
          [smtp.example.com\]:587 [[USER@example.com:PASS]](mailto:USER@example.
  ↪com:PASS)
  ```

You need to give appropriate permissions:

```
chmod 400 /etc/postfix/sasl_passwd
```

and map configuration to database:

```
postmap /etc/postfix/sasl_passwd
```

next you need to generate a ca cert file:

```
cat /etc/ssl/certs/Example\_Server\_CA.pem | tee -a etc/postfix/cacert.pem
```

And finally, you need to restart postfix

```
/etc/init.d/postfix restart
```

API

18.1 Kibana API

The Kibana dashboard import/export APIs allow people to import dashboards along with all of their corresponding saved objects such as visualizations, saved searches, and index patterns.

18.1.1 Kibana Import API

Request:

```
POST /api/kibana/dashboards/import
```

Query Parameters:

- `force` (optional)

  (boolean) Overwrite any existing objects on id conflict

- `exclude` (optional)

  (array) Saved object types that should not be imported

  Example:

  ```
  curl -X POST "https://user:password@localhost:5601POST api/kibana/dashboards/
  ↪import?exclude=index-pattern"
  ```

18.1.2 Kibana Export API

Request:

```
GET /api/kibana/dashboards/export
```

Query Parameters

- `dashboard` (required)

  (array|string) The id(s) of the dashboard(s) to export

  Example:

```
curl -k -XPOST "https://user:password@localhost:443/api/kibana/dashboards/import?
↪force=true&exclude=index-pattern" -H 'kbn-xsrf: true' -H 'Content-Type:␣
↪application/json' -d@dashboard.json
```

18.2 Elasticsearch API

The Elasticsearch has a typical REST API and data is received in JSON format after the HTTP protocol. By default the tcp/9200 port is used to communicate with the Elasticsearch API. For purposes of examples, communication with the Elasticsearch API will be carried out using the *curl* application.

Program syntax:

```
# curl -XGET -u login:password '127.0.0.1:9200'
```

Available methods:

- PUT - sends data to the server;
- POST - sends a request to the server for a change;
- DELETE - deletes the index / document;
- GET - gets information about the index /document;
- HEAD - is used to check if the index / document exists.

Avilable APIs by roles:

- Index API - manages indexes;
- Document API - manges documnets;
- Cluster API - manage the cluster;
- Search API - is userd to search for data.

18.3 Elasticsearch Index API

The indices APIs are used to manage individual indices, index settings, aliases, mappings, and index templates.

18.3.1 Adding Index

Adding Index - autormatic method:

```
curl -XPUT -u login:password '127.0.0.1:9200/twitter/tweet/1?pretty=true' -d'{
    "user" : "elk01",
    "post_date" : "2017-09-05T10:00:00",
    "message" : "tests auto index generation"
    }'
```

You should see the output:

```
{
"_index" : "twitter",
  "_type" : "tweet",
  "_id" : "1",
  "_version" : 1,
  "_shards" : {
    "total" : 2,
    "successful" : 1,
    "failed" : 0
  },
  "created" : true
}
```

The parameter `action.auto_create_index` must be set on `true`.

Adding Index – manual method:

> • settings the number of shards and replicas:

```
curl -XPUT -u login:password '127.0.0.1:9200/twitter2?pretty=true' -d'{
        "settings" : {
        "number_of_shards" : 1,
        "number_of_replicas" : 1
        }
 }'
```

You should see the output:

```
{
  "acknowledged" : true
}
```

> • command for manual index generation:

```
curl -XPUT -u login:password '127.0.0.1:9200/twitter2/tweet/1?pretty=true' -d'{
                                "user" : "elk01",
                                "post_date" : "2017-09-05T10:00:00",
                                "message" : "tests manual index generation"
                        }'
```

You should see the output:

```
{
  "_index" : "twitter2",
  "_type" : "tweet",
  "_id" : "1",
  "_version" : 1,
  "_shards" : {
    "total" : 2,
    "successful" : 1,
    "failed" : 0
  },
  "created" : true
}
```

18.3.2 Delete Index

Delete Index - to delete *twitter* index you need use the following command:

```
curl -XDELETE -u login:password '127.0.0.1:9200/twitter?pretty=true'
```

The delete index API can also be applied to more than one index, by either using a comma separated list, or on all indice by using _all or * as index:

```
curl -XDELETE -u login:password '127.0.0.1:9200/twitter*?pretty=true'
```

To allowing to delete indices via wildcards set `action.destructive_requires_name` setting in the config to `false`.

18.3.3 API useful commands

- get information about Replicas and Shards:

```
curl -XGET -u login:password '127.0.0.1:9200/twitter/_settings?pretty=true'
```

```
curl -XGET -u login:password '127.0.0.1:9200/twitter2/_settings?pretty=true'
```

- get information about mapping and alias in the index:

```
curl -XGET -u login:password '127.0.0.1:9200/twitter/_mappings?pretty=true'
```

```
curl -XGET -u login:password '127.0.0.1:9200/twitter/_aliases?pretty=true'
```

- get all information about the index:

```
curl -XGET -u login:password '127.0.0.1:9200/twitter?pretty=true'
```

    – checking does the index exist:

```
curl -XGET -u login:password '127.0.0.1:9200/twitter?pretty=true'
```

- close the index:

```
curl -XPOST -u login:password '127.0.0.1:9200/twitter/_close?pretty=true'
```

- open the index:

```
curl -XPOST -u login:password '127.0.0.1:9200/twitter/_open?pretty=true'
```

- get the status of all indexes:

```
curl -XGET -u login:password '127.0.0.1:9200/_cat/indices?v'
```

- get the status of one specific index:

```
curl -XGET -u login:password '127.0.0.1:9200/_cat/indices/twitter?v'
```

- display how much memory is used by the indexes:

```
curl -XGET -u login:password '127.0.0.1:9200/_cat/indices?v&h=i,tm&s=tm:desc'
```

- display details of the shards:

```bash
    ```bash
```

curl -XGET -u login:password '127.0.0.1:9200/_cat/shards?v'

18.4 Elasticsearch Document API

18.4.1 Create Document

- create a document with a specify ID:

```
curl -XPUT -u login:password '127.0.0.1:9200/twitter/tweet/1?pretty=true' -d'{
                        "user" : "lab1",
                        "post_date" : "2017-08-25T10:00:00",
                        "message" : "testuje Elasticsearch"
                }'
```

You should see the output:

```
{
  "_index" : "twitter",
  "_type" : "tweet",
  "_id" : "1",
  "_version" : 1,
  "_shards" : {
    "total" : 2,
    "successful" : 1,
    "failed" : 0
  },
  "created" : true
}
```

- creating a document with an automatically generated ID: (note: PUT-> POST):

```bash
        ```bash
```

curl -XPOST -u login:password '127.0.0.1:9200/twitter/tweet?pretty=true' -d'{ "user" : "lab1", "post_date" : "2017-08-25T10:10:00", "message" : "testuje automatyczne generowanie ID" }' "user" : "lab1", "post_date" : "2017-08-25T10:10:00", "message" : "testuje automatyczne generowanie ID" }'

You should see the output:

```
{
  "_index" : "twitter",
  "_type" : "tweet",
  "_id" : "AV49sTlM8NzerkV9qJfh",
  "_version" : 1,
  "_shards" : {
    "total" : 2,
    "successful" : 1,
    "failed" : 0
  },
  "created" : true
}
```

18.4.2 Delete Document

- delete a document by ID:

```
curl -XDELETE -u login:password '127.0.0.1:9200/twitter/tweet/1?pretty=true'
```

```
curl -XDELETE -u login:password '127.0.0.1:9200/twitter/tweet/
↪AV49sTlM8NzerkV9qJfh?pretty=true'
```

- delete a document using a wildcard:

```
curl -XDELETE -u login:password '127.0.0.1:9200/twitter/tweet/1*?pretty=true'
```

(parametr: action.destructive_requires_name must be set to false)

18.4.3 Useful commands

- get information about the document:

```
curl -XGET -u login:password '127.0.0.1:9200/twitter/tweet/1?pretty=true'
```

You should see the output:

```
{
  "_index" : "twitter",
  "_type" : "tweet",
  "_id" : "1",
  "_version" : 1,
  "found" : true,
  "_source" : {
    "user" : "lab1",
    "post_date" : "2017-08-25T10:00:00",
    "message" : "testuje Elasticsearch"
  }
}
```

- get the source of the document:

```
curl -XGET -u login:password '127.0.0.1:9200/twitter/tweet/1/_source?
↪pretty=true'
```

You should see the output:

```
{
  "user" : "lab1",
  "post_date" : "2017-08-25T10:00:00",
  "message" : "test of Elasticsearch"
}
```

- get information about all documents in the index:

```
        curl -XGET -u login:password '127.0.0.1:9200/twitter*/_search?q=*&
↪pretty=true'
```

You should see the output:

```
{
  "took" : 7,
  "timed_out" : false,
  "_shards" : {
    "total" : 10,
    "successful" : 10,
    "failed" : 0
  },
  "hits" : {
    "total" : 3,
    "max_score" : 1.0,
    "hits" : [ {
      "_index" : "twitter",
      "_type" : "tweet",
      "_id" : "AV49sTlM8NzerkV9qJfh",
      "_score" : 1.0,
      "_source" : {
        "user" : "lab1",
        "post_date" : "2017-08-25T10:10:00",
        "message" : "auto generated ID"
      }
    }, {
      "_index" : "twitter",
      "_type" : "tweet",
      "_id" : "1",
      "_score" : 1.0,
      "_source" : {
        "user" : "lab1",
        "post_date" : "2017-08-25T10:00:00",
        "message" : "Elasticsearch test"
      }
    }, {
      "_index" : "twitter2",
      "_type" : "tweet",
      "_id" : "1",
      "_score" : 1.0,
      "_source" : {
        "user" : "elk01",
        "post_date" : "2017-09-05T10:00:00",
        "message" : "manual index created test"
      }
    } ]
  }
}
```

- the sum of all documents in a specified index:

```
curl -XGET -u login:password '127.0.0.1:9200/_cat/count/twitter?v'
```

```
You should see the output:

epoch                  timestamp count
1504281400        17:56:40     2
```

- the sum of all document in Elasticsearch database:

```
```bash
curl -XGET -u login:password '127.0.0.1:9200/_cat/count?v'
```

```
      You should see the output:
```

```
   epoch                    timestamp count
   1504281518          17:58:38    493658
```

18.5 Elasticsearch Cluster API

18.5.1 Useful commands

- information about the cluster state:

```
curl -XGET -u login:password '127.0.0.1:9200/_cluster/health?pretty=true'
```

- information about the role and load of nodes in the cluster:

```
curl -XGET -u login:password '127.0.0.1:9200/_cat/nodes?v'
```

- information about the available and used place on the cluster nodes:

```
curl -XGET -u login:password '127.0.0.1:9200/_cat/allocation?v'
```

- information which node is currently in the master role:

```
```bash
```

curl -XGET -u login:password '127.0.0.1:9200/_cat/master?v' ```

- information abut currently performed operations by the cluster:

  - ```
    curl -XGET -u login:password '127.0.0.1:9200/_cat/pending_tasks?v'
    ```

  information on revoceries / transferred indices:

  ```
  curl -XGET -u login:password '127.0.0.1:9200/_cat/recovery?v'
  ```

- information about shards in a cluster:

```
curl -XGET -u login:password '127.0.0.1:9200/_cat/shards?v'
```

- detailed inforamtion about the cluster:

```bash
```

curl -XGET -u login:password '127.0.0.1:9200/_cluster/stats?human&pretty' ```

- detailed information about the nodes:

```
curl -XGET -u login:password '127.0.0.1:9200/_nodes/stats?human&pretty'
```

18.6 Elasticsearch Search API

18.6.1 Useful commands

- searching for documents by the string:

```
curl -XPOST -u login:password '127.0.0.1:9200/twitter*/tweet/_search?pretty=true' -d '
→{
                        "query": {
                                "bool" : {
                                        "must" : {
                                        "query_string" : {
                                                "query" : "test"
                                        }
                                    }
                                }
                        }
                }'
```

- searching for document by the string and filtering:

```bash
```

curl -XPOST -u login:password '127.0.0.1:9200/twitter*/tweet/_search?pretty=true' -d'{

```
                "query": {
                        "bool" : {
                                "must" : {
                                "query_string" : {
                                        "query" : "testuje"
                                }
                            },
                                "filter" : {
                                    "term" : { "user" : "lab1" }
                                }
                        }
                }
        }'
```

- simple search in a specific field (in this case user) uri query:

```
curl -XGET -u login:password '127.0.0.1:9200/twitter*/_search?q=user:lab1&
→pretty=true'
```

- simple search in a specific field:

```bash
```

curl -XPOST -u login:password '127.0.0.1:9200/twitter*/_search?pretty=true' -d '{

```
                            "query" : {
                                    "term" : { "user" : "lab1" }
                    }
        }'
```

18.7 Elasticsearch - Mapping, Fielddata and Templates

Mapping is a collection of fields along with a specific data type Fielddata is the field in which the data is stored (requires a specific type - string, float) Template is a template based on which fielddata will be created in a given index.

18.7.1 Useful commands

- Information on all set mappings:

```
curl -XGET -u login:password '127.0.0.1:9200/_mapping?pretty=true'
```

- Information about all mappings set in the index:

```
curl -XGET -u login:password '127.0.0.1:9200/twitter/_mapping/*?pretty=true'
```

- Information about the type of a specific field:

```
curl -XGET -u login:password '127.0.0.1:9200/twitter/_mapping/field/message*?
↪pretty=true'
```

- Information on all set templates:

```bash
curl  -XGET -u login:password '127.0.0.1:9200/_template/*?pretty=true'
```

```
        ### Create - Mapping / Fielddata
```

- Create - Mapping / Fielddata - It creates index twitter-float and the tweet message field sets to float:

```
curl -XPUT -u login:password '127.0.0.1:9200/twitter-float?pretty=true' -d '{
  "mappings": {
    "tweet": {
      "properties": {
        "message": {
          "type": "float"
        }
      }
    }
```

(continues on next page)

```
  }
}'

curl -XGET -u login:password '127.0.0.1:9200/twitter-float/_mapping/field/message?
↪pretty=true'
```

18.7.2 Create Template

- Create Template:

```
  ```bash
```

curl -XPUT -u login:password '127.0.0.1:9200/_template/template_1' -d'{ "template" : "twitter4", "order" : 0, "settings" : { "number_of_shards" : 2 } }'

```bash
curl -XPOST -u login:password '127.0.0.1:9200/twitter4/tweet?pretty=true' -d'{
        "user" : "lab1",
        "post_date" : "2017-08-25T10:10:00",
        "message" : "test of ID generation"
}'
```

```
curl -XGET -u login:password '127.0.0.1:9200/twitter4/_settings?pretty=true'
```

- Create Template2 - Sets the mapping template for all new indexes specifying that the tweet data, in the field called message, should be of the "string" type:

```
curl -XPUT -u login:password '127.0.0.1:9200/_template/template_2' -d'{
  "template" : "*",
  "mappings": {
    "tweet": {
      "properties": {
        "message": {
          "type": "string"
        }
      }
    }
  }
}'
```

18.7.3 Delete Mapping

- Delete Mapping - Deleting a specific index mapping (no possibility to delete - you need to index):

```
  ```bash
  curl -XDELETE -u login:password '127.0.0.1:9200/twitter2'
```

18.7.4 Delete Template

- Delete Template:

```bash
   ```bash
   curl  -XDELETE -u login:password '127.0.0.1:9200/_template/template_1?
↪pretty=true'
```

18.8 AI Module API

18.8.1 Services

The intelligence module has implemented services that allow you to create, modify, delete, execute and read definitions of AI rules.

18.8.2 List rules

The list service returns a list of AI rules definitions stored in the system.

Method: GET URL:

```
https://<host>:<port>/api/ai/list?pretty
```

where:

```
host        -        kibana host address
port        -        kibana port
?pretty     -         optional json format parameter
```

Curl:

```
curl -XGET 'https://localhost:5601/api/ai/list?pretty' -u <user>:<password> -k
```

Result: Array of JSON documents:

```
| Field                              | Value                                      ␣
↪                          | Screen field (description)␣
↪         |
|-------------------------------     |--------------------------------------------
↪-----------------------------------     |---------------------------
↪         |
| _source.algorithm_type             | GMA, GMAL, LRS, LRST, RFRS, SMAL, SMA, TL ␣
↪                          | Algorithm.                            ␣
↪         |
| _source.model_name                 | Not empty string.                          ␣
↪                          | AI Rule Name.                         ␣
↪         |
| _source.search                     | Search id.                                 ␣
↪                          | Choose search.                        ␣
↪         |
| _source.label_field.field          |                                            ␣
↪                          | Feature to analyse.                   ␣
↪         |
```

```
| _source.max_probes                    | Integer value                       ␣
↪                                       | Max probes                          ␣
↪        |
| _source.time_frame                    | 1 minute, 5 minutes, 15 minutes, 30␣
↪minutes, 1 hour, 1 day, 1 week, 30 day, 365 day      | Time frame          ␣
↪        |
| _source.value_type                    | min, max, avg, count                ␣
↪                                       | Value type                          ␣
↪        |
| _source.max_predictions               | Integer value                       ␣
↪                                       | Max predictions                     ␣
↪        |
| _source.threshold                     | Integer value                       ␣
↪                                       | Threshold                           ␣
↪        |
| _source.automatic_cron                | Cron format string                  ␣
↪                                       | Automatic cycle                     ␣
↪        |
| _source.automatic_enable              | true/false                          ␣
↪                                       | Enable                              ␣
↪        |
| _source.automatic                     | true/false                          ␣
↪                                       | Automatic                           ␣
↪        |
| _source.start_date                    | YYYY-MM-DD HH:mm or now             ␣
↪                                       | Start date                          ␣
↪        |
| _source.multiply_by_values            | Array of string values              ␣
↪                                       | Multiply by values                  ␣
↪        |
| _source.multiply_by_field             | None or full field name eg.: system.cpu ␣
↪                                       | Multiply by field                   ␣
↪        |
| _source.selectedroles                 | Array of roles name                 ␣
↪                                       | Role                                ␣
↪        |
| _source.last_execute_timestamp        |                                     ␣
↪                                       | Last execute                        ␣
↪        |
```

Not screen fields:

```
| _index                                |        | Elasticsearch index name.      ␣
↪              |
|------------------------------         |---     |--------------------------------
↪----          |
| _type                                 |        | Elasticsearch document type.   ␣
↪              |
| _id                                   |        | Elasticsearch document id.     ␣
↪              |
| _source.preparation_date              |        | Document preparation date.     ␣
↪              |
| _source.machine_state_uid             |        | AI rule machine state uid.     ␣
↪              |
| _source.path_to_logs                  |        | Path to ai machine logs.       ␣
↪              |
| _source.path_to_machine_state         |        | Path to ai machine state files.␣
↪              |
```

```
| _source.searchSourceJSON            |                     | Query string.              ␣
↪            |
| _source.processing_time             |                     | Process operation time.     ␣
↪            |
| _source.last_execute_mili           |                     | Last executed time in␣
↪milliseconds.          |
| _source.pid                         |                     | Process pid if ai rule is␣
↪running.            |
| _source.exit_code                   |                     | Last executed process exit code.
↪            |
```

18.8.3 Show rules

The show service returns a document of AI rule definition by id.

Method: GET URL: https://:/api/ai/show/?pretty

where:

```
host        -        kibana host address
port        -        kibana port
id          -        ai rule document id
?pretty        -        optional json format parameter
```

Curl:

```
curl -XGET 'https://localhost:5601/api/ai/show/ea9384857de1f493fd84dabb6dfb99ce?pretty
↪' -u <user>:<password> -k
```

Result JSON document:

```
| Field                               | Value                              ␣
↪                                    | Screen field (description)␣
↪        |
|------------------------------       |------------------------------------------
↪------------------------------------        |---------------------------
↪        |
| _source.algorithm_type              | GMA, GMAL, LRS, LRST, RFRS, SMAL, SMA, TL ␣
↪                                    | Algorithm.                 ␣
↪        |
| _source.model_name                  | Not empty string.                  ␣
↪                                    | AI Rule Name.              ␣
↪        |
| _source.search                      | Search id.                         ␣
↪                                    | Choose search.             ␣
↪        |
| _source.label_field.field           |                                    ␣
↪                                    | Feature to analyse.        ␣
↪        |
| _source.max_probes                  | Integer value                      ␣
↪                                    | Max probes                 ␣
↪        |
| _source.time_frame                  | 1 minute, 5 minutes, 15 minutes, 30␣
↪minutes, 1 hour, 1 day, 1 week, 30 day, 365 day        | Time frame                 ␣
↪        |
| _source.value_type                  | min, max, avg, count               ␣
↪                                    | Value type
↪        |
```

```
| _source.max_predictions                   | Integer value                          ␣
↪                        | Max predictions                       ␣
↪          |
| _source.threshold                         | Integer value                          ␣
↪                        | Threshold                             ␣
↪          |
| _source.automatic_cron                    | Cron format string                     ␣
↪                        | Automatic cycle                       ␣
↪          |
| _source.automatic_enable                  | true/false                             ␣
↪                        | Enable                                ␣
↪          |
| _source.automatic                         | true/false                             ␣
↪                        | Automatic                             ␣
↪          |
| _source.start_date                        | YYYY-MM-DD HH:mm or now                 ␣
↪                        | Start date                            ␣
↪          |
| _source.multiply_by_values                | Array of string values                 ␣
↪                        | Multiply by values                    ␣
↪          |
| _source.multiply_by_field                 | None or full field name eg.: system.cpu ␣
↪                        | Multiply by field                     ␣
↪          |
| _source.selectedroles                     | Array of roles name                    ␣
↪                        | Role                                  ␣
↪          |
| _source.last_execute_timestamp            |                                        ␣
↪                        | Last execute                          ␣
↪          |
```

Not screen fields

```
| _index                     |        | Elasticsearch index name.       ␣
↪            |
|----------------------------       |---     |-------------------------------
↪----        |
| _type                      |        | Elasticsearch document type.    ␣
↪            |
| _id                        |        | Elasticsearch document id.      ␣
↪            |
| _source.preparation_date   |        | Document preparation date.      ␣
↪            |
| _source.machine_state_uid  |        | AI rule machine state uid.      ␣
↪            |
| _source.path_to_logs       |        | Path to ai machine logs.        ␣
↪            |
| _source.path_to_machine_state |     | Path to ai machine state files.␣
↪            |
| _source.searchSourceJSON   |        | Query string.                   ␣
↪            |
| _source.processing_time    |        | Process operation time.         ␣
↪            |
| _source.last_execute_mili  |        | Last executed time in␣
↪milliseconds.           |
| _source.pid                |        | Process pid if ai rule is␣
↪running.           |
```

```
| _source.exit_code                  |            | Last executed process exit code.
↪          |
```

18.8.4 Create rules

The create service adds a new document with the AI rule definition.

Method: PUT

URL:

```
https://<host>:<port>/api/ai/create
```

where:

```
host       -         kibana host address
port       -         kibana port
body       -         JSON with definition of ai rule
```

Curl:

```
curl -XPUT 'https://localhost:5601/api/ai/create' -u <user>:<password> -k -H "kbn-
↪version: 6.2.4" -H 'Content-type: application/json' -d' {"algorithm_type":"TL",
↪"model_name":"test","search":"search:6c226420-3b26-11e9-a1c0-4175602ff5d0","label_
↪field":{"field":"system.cpu.idle.pct"},"max_probes":100,"time_frame":"1 day","value_
↪type":"avg","max_predictions":10,"threshold":-1,"automatic_cron":"*/5 * * * *",
↪"automatic_enable":true,"automatic_flag":true,"start_date":"now","multiply_by_values
↪":[],"multiply_by_field":"none","selectedroles":["test"]}'
```

Validation:

```
| Field                  | Values                                                       ␣
↪                           |
|---------------         |------------------------------------------------------------
↪------------------------        |
| algorithm_type         | GMA, GMAL, LRS, LRST, RFRS, SMAL, SMA, TL                    ␣
↪                           |
| value_type             | min, max, avg, count                                         ␣
↪                           |
| time_frame             | 1 minute, 5 minutes, 15 minutes, 30 minutes, 1 hour, 1 day,
↪ 1 week, 30 day, 365 day        |
```

Body JSON description:

```
| Field                        | Mandatory              | Value                       ␣
↪                                            | Screen field      ␣
↪            |
|-------------------           |------------------       |----------------------------
↪----------------------------------------------       |------------------
↪--        |
| algorithm_type               | Yes                     | GMA, GMAL, LRS, LRST, RFRS,␣
↪SMAL, SMA, TL                                          | Algorithm.        ␣
↪            |
| model_name                   | Yes                     | Not empty string.         ␣
↪                                            | AI Rule Name.     ␣
↪            |
```

```
| search                     | Yes                         | Search id.                  ␣
↪                            |                             | Choose search.              ␣
↪             |
| label_field.field          | Yes                         |                             ␣
↪                            |                             | Feature to␣
↪analyse.          |
| max_probes                 | Yes                         | Integer value               ␣
↪                            |                             | Max probes                  ␣
↪             |
| time_frame                 | Yes                         | 1 minute, 5 minutes, 15␣
↪minutes, 30 minutes, 1 hour, 1 day, 1 week, 30 day, 365 day       | Time frame  ␣
↪             |
| value_type                 | Yes                         | min, max, avg, count        ␣
↪                            |                             | Value type                  ␣
↪             |
| max_predictions            | Yes                         | Integer value               ␣
↪                            |                             | Max predictions             ␣
↪             |
| threshold                  | No (default -1)             | Integer value               ␣
↪                            |                             | Threshold                   ␣
↪             |
| automatic_cron             | Yes                         | Cron format string          ␣
↪                            |                             | Automatic cycle             ␣
↪             |
| Automatic_enable           | Yes                         | true/false                  ␣
↪                            |                             | Enable                      ␣
↪             |
| automatic                  | Yes                         | true/false                  ␣
↪                            |                             | Automatic                   ␣
↪             |
| start_date                 | No (default now)            | YYYY-MM-DD HH:mm or now      ␣
↪                            |                             | Start date                  ␣
↪             |
| multiply_by_values         | Yes                         | Array of string values      ␣
↪                            |                             | Multiply by␣
↪values       |
| multiply_by_field          | Yes                         | None or full field name eg. ␣
↪: system.cpu                |                             | Multiply by␣
↪field       |
| selectedroles              | No                          | Array of roles name         ␣
↪                            |                             | Role                        ␣
↪             |
```

Result:

JSON document with fields:

```
status      -       true if ok
id          -       id of changed document
message     -        error message
```

18.8.5 Update rules

The update service changes the document with the AI rule definition.

Method:POST

URL:

```
https://<host>:<port>/api/ai/update/<id>
```

where:

```
host        -         kibana host address
port        -         kibana port
id          -        ai rule document id
body        -         JSON with definition of ai rule
```

Curl:

```
curl -XPOST 'https://localhost:5601/api/ai/update/ea9384857de1f493fd84dabb6dfb99ce' -
↪u <user>:<password> -k -H "kbn-version: 6.2.4" -H 'Content-type: application/json' -
↪d'
{"algorithm_type":"TL","search":"search:6c226420-3b26-11e9-a1c0-4175602ff5d0","label_
↪field":{"field":"system.cpu.idle.pct"},"max_probes":100,"time_frame":"1 day","value_
↪type":"avg","max_predictions":100,"threshold":-1,"automatic_cron":"*/5 * * * *",
↪"automatic_enable":true,"automatic_flag":true,"start_date":"now","multiply_by_values
↪":[],"multiply_by_field":"none","selectedroles":["test"]}
```

Validation:

```
| Field                   | Values                                              ␣
↪                        |
|---------------          |----------------------------------------------------
↪------------------------         |
| algorithm_type          | GMA, GMAL, LRS, LRST, RFRS, SMAL, SMA, TL           ␣
↪                        |
| value_type              | min, max, avg, count                                ␣
↪                        |
| time_frame              | 1 minute, 5 minutes, 15 minutes, 30 minutes, 1 hour, 1 day,
↪ 1 week, 30 day, 365 day       |
```

Body JSON description:

```
| Field                     | Mandatory          | Value                        ␣
↪                              | Screen field       ␣
↪          |
|-------------------        |-----------------   |----------------------------
↪------------------------------------------   |-----------------
↪--       |
| algorithm_type            | Yes                | GMA, GMAL, LRS, LRST, RFRS,␣
↪SMAL, SMA, TL                                | Algorithm.         ␣
↪          |
| model_name                | Yes                | Not empty string.          ␣
↪                              | AI Rule Name.      ␣
↪          |
| search                    | Yes                | Search id.                 ␣
↪                              | Choose search.     ␣
↪          |
| label_field.field         | Yes                |                              ␣
↪                              | Feature to␣
↪analyse.          |
| max_probes                | Yes                | Integer value                ␣
↪                              | Max probes         ␣
↪          |
```

(continues on next page)

```
| time_frame                      | Yes                            | 1 minute, 5 minutes, 15␣
↪minutes, 30 minutes, 1 hour, 1 day, 1 week, 30 day, 365 day        | Time frame       ␣
↪             |
| value_type                      | Yes                            | min, max, avg, count   ␣
↪                                          | Value type       ␣
↪             |
| max_predictions                 | Yes                            | Integer value          ␣
↪                                          | Max predictions  ␣
↪             |
| threshold                       | No (default -1)                | Integer value          ␣
↪                                          | Threshold        ␣
↪             |
| automatic_cron                  | Yes                            | Cron format string     ␣
↪                                          | Automatic cycle  ␣
↪             |
| Automatic_enable                | Yes                            | true/false             ␣
↪                                          | Enable           ␣
↪             |
| automatic                       | Yes                            | true/false             ␣
↪                                          | Automatic        ␣
↪             |
| start_date                      | No (default now)               | YYYY-MM-DD HH:mm or now ␣
↪                                          | Start date       ␣
↪             |
| multiply_by_values              | Yes                            | Array of string values ␣
↪                                          | Multiply by␣
↪values          |
| multiply_by_field               | Yes                            | None or full field name eg.
↪: system.cpu                                          | Multiply by␣
↪field           |
| selectedroles                   | No                             | Array of roles name    ␣
↪                                          | Role             ␣
↪             |
```

Result:

JSON document with fields:

```
    status      -        true if ok
    id          -        id of changed document
    message     -         error message
```

Run:

The run service executes a document of AI rule definition by id.

Method: GET

URL:

```
https://<host>:<port>/api/ai/run/<id>
```

where:

```
    host        -        kibana host address
    port        -        kibana port
    id          -        ai rule document id
```

Curl:

```
      curl -XGET 'https://localhost:5601/api/ai/run/ea9384857de1f493fd84dabb6dfb99ce
↪' -u <user>:<password> -k
```

Result:

JSON document with fields:

```
    status       -          true if ok
    id        -          id of executed document
    message      -          message
```

18.8.6 Delete rules

The delete service removes a document of AI rule definition by id.

Method: DELETE

URL:

```
https://<host>:<port>/api/ai/delete/<id>
```

where:

```
    host       -         kibana host address
    port       -         kibana port
    id       -        ai rule document id
```

Curl:

```
curl -XDELETE 'https://localhost:5601/api/ai/delete/ea9384857de1f493fd84dabb6dfb99ce'
↪-u <user>:<password> -k -H "kbn-version: 6.2.4"
```

Result:

JSON document with fields:

```
status       -         true if ok
id       -        id of executed document
message      -         message
```

18.9 Alert module API

18.9.1 Create Alert Rule

Method: POST

```
URL: /api/admin/alertrules
```

Body:

In the body of call, you must pass the JSON object with the full definition of the rule document:

```
| Name                  | Description                                                  ␣
↪                                                                                 ␣
↪                                  |
|---------------------|-------------------------------------------------------------
↪-----------------------------------------------------------------------------
↪------------------------------|
| id                    | Document ID in Elasticsearch                                ␣
↪                                                                                 ␣
↪                                  |
| alertrulename         | Rule name (the Name field from the Create Alert tab  the␣
↪name must be the same as the alert name)                                         ␣
↪                                  |
| alertruleindexpattern | Index pattern (Index pattern field from the Create Alert␣
↪tab)                                                                             ␣
↪                                  |
| selectedroles         | Array of roles that have rights to this rule (Roles field␣
↪from the Create Alert tab)                                                       ␣
↪                                  |
| alertruletype         | Alert rule type (Type field from the Create Alert tab)   ␣
↪                                                                                 ␣
↪                                  |
| alertrulemethod       | Type of alert method (Alert method field from the Create␣
↪Alert tab)                                                                       ␣
↪                                  |
| alertrulemethoddata   | Data for the type of alert (field Email address if␣
↪alertrulemethod is email  Path to script / command if alertrulemethod is command ␣
↪and empty value if alertrulemethod is none) |
| alertrule_any         | Alert script (the Any field from the Create Alert tab)   ␣
↪                                                                                 ␣
↪                                  |
| alertruleimportance   | Importance of the rule (Rule importance box from the Create␣
↪Alert tab)                                                                       ␣
↪                                  |
| alertruleriskkey      | Field for risk calculation (field from the index indicated␣
↪by alertruleindexpattern according to which the risk will be counted  Risk key␣
↪field from the Create Alert tab)       |
| alertruleplaybooks    | Playbook table (document IDs) attached to the alert␣
↪(Playbooks field from the Create Alert tab)                                      ␣
↪                                       |
| enable                | Value Y or N depending on whether we enable or disable the␣
↪rule                                                                             ␣
↪                                  |
| authenticator         | Constant value index                                     ␣
↪                                                                                 ␣
↪                                  |
```

Result OK:

```
"Successfully created rule!!"
```

or if fault, error message.

Example:

```
curl -XPOST 'https://localhost:5601/api/admin/alertrules' -u user:passowrd -k -H "kbn-
↪version: 6.2.4" -H 'Content-type: application/json' -d'
{
        "id":"test_enable_rest",
```

(continues on next page)

```
        "alertrulename":"test enable rest",
        "alertruleindexpattern":"m*",
        "selectedroles":"",
        "alertruletype":"frequency",
        "alertrulemethod":"email",
        "alertrulemethoddata":"ala@local",
        "alertrule_any":"# (Required, frequency specific)\n# Alert when this many␣
↪documents matching the query occur within a timeframe\nnum_events: 5\n\n# (Required,
↪ frequency specific)\n# num_events must occur within this amount of time to trigger␣
↪an alert\ntimeframe:\n  minutes: 2\n\n# (Required)\n# A list of Elasticsearch␣
↪filters used for find events\n# These filters are joined with AND and nested in a␣
↪filtered query\n# For more info: http://www.elasticsearch.org/guide/en/
↪elasticsearch/reference/current/query-dsl.html\nfilter:\n- term:\n    some_field: \
↪"some_value\"\n\n# (Optional, change specific)\n# If true, Alert will poll␣
↪Elasticsearch using the count api, and not download all of the matching documents.␣
↪This is useful is you care only about numbers and not the actual data. It should␣
↪also be used if you expect a large number of query hits, in the order of tens of␣
↪thousands or more. doc_type must be set to use this.\n#use_count_query:\n\n#␣
↪ (Optional, change specific)\n# Specify the _type of document to search for. This␣
↪must be present if use_count_query or use_terms_query is set.\n#doc_type:\n\n#␣
↪ (Optional, change specific)\n# If true, Alert will make an aggregation query␣
↪against Elasticsearch to get counts of documents matching each unique value of␣
↪query_key. This must be used with query_key and doc_type. This will only return a␣
↪maximum of terms_size, default 50, unique terms.\n#use_terms_query:\n# (Optional,␣
↪change specific)\n# When used with use_terms_query, this is the maximum number of␣
↪terms returned per query. Default is 50.\n#terms_size:\n\n# (Optional, change␣
↪specific)\n# Counts of documents will be stored independently for each value of␣
↪query_key. Only num_events documents, all with the same value of query_key, will␣
↪trigger an alert.\n#query_key:\n\n# (Optional, change specific)\n# Will attach all␣
↪the related events to the event that triggered the frequency alert. For example in␣
↪an alert triggered with num_events: 3, the 3rd event will trigger the alert on␣
↪itself and add the other 2 events in a key named related_events that can be␣
↪accessed in the alerter.\n#attach_related:",
        "alertruleplaybooks":[],
        "alertruleimportance":50,
        "alertruleriskkey":"beat.hostname",
        "enable":"Y",
        "authenticator":"index"
}
'
```

18.9.2 Save Alert Rules

Method: POST

URL:

```
/api/admin/saverules
```

Body:

In the body of call, you must pass the JSON object:

```
'authenticator'
```

Constant value `index`

Result:

```
"Files created"
```

or if fault, error message.

Example:

```
curl -XPOST 'https://localhost:5601/api/admin/saverules' -u user:password -k -H "kbn-
→version: 6.2.4" -H 'Content-type: application/json' -d'
        {
                "authenticator":"index"
        }
'
```

18.10 Reports module API

18.10.1 Create new task

CURL query to create a new csv report:

```
curl -k "https://localhost:5601/api/taskmanagement/export" -XPOST -H 'kbn-xsrf: true'␣
→-H 'Content-Type: application/json;charset=utf-8' -u USER:PASSWORD -d '{
  "indexpath": "audit",
  "query": "*",
  "fields": [
    "@timestamp",
    "method",
    "operation",
    "request",
    "username"
  ],
  "initiatedUser": "logserver ",
  "fromDate": "2019-09-18T00:00:00",
  "toDate": "2019-09-19T00:00:00",
  "timeCriteriaField": "@timestamp",
  "export_type": "csv",
  "export_format": "csv",
  "role": ""
}'
```

Answer:

```
{"taskId":"1568890625355-cbbe16e1-12ac-b53c-158e-e0919338953c"}
```

bash### Checking the status of the task ###

```
curl -k -XGET -u USER:PASSWORD https://localhost:5601/api/taskmanagement/export/
→1568890625355-cbbe16e1-12ac-b53c-158e-e0919338953
```

Answer:

- In progress:

  ```bash
  {"taskId":"1568890766279-56667dc8-6bd4-3f42-1773-08722b623ec1","status":
  →"Processing"}
  ```

* Done:

```
    {"taskId":"1568890625355-cbbe16e1-12ac-b53c-158e-e0919338953c","status":
→"Complete","download":"http://localhost:5601/api/taskmanagement/export/
→1568890625355-cbbe16e1-12ac-b53c-158e-e0919338953c/download"}
```

* Error during execution:

```
```bash
{"taskId":"1568890794564-120f0549-921f-4459-3114-3ea3f6e861b8","status":"Error␣
→Occured"}
```

18.10.2 Downloading results

```
    curl -k -XGET -u USER:PASSWORD https://localhost:5601/api/taskmanagement/
→export/1568890625355-cbbe16e1-12ac-b53c-158e-e0919338953c/download > /tmp/audit_
→report.csv
```

18.11 License module API

You can check the status of the ITRS Log Analytics license via the API

Method: GET

Curl:

```
curl -u $USER:$PASSWORD -X GET http://localhost:9200/license
```

Result:

```
{"status":200,"nodes":"10","indices":"[*]","customerName":"example","issuedOn":"2019-
→05-27T12:16:16.174326700","validity":"100","documents":"","version":"6.1.6"}
```

CHAPTER 19

Logstash

The ITRS Log Analytics use Logstash service to dynamically unify data from disparate sources and normalize the data into destination of your choose. A Logstash pipeline has two required elements, *input* and *output*, and one optional element *filter*. The input plugins consume data from a source, the filter plugins modify the data as you specify, and the output plugins write the data to a destination. The default location of the Logstash plugin files is: */etc/logstash/conf.d/*. This location contain following ITRS Log Analytics

ITRS Log Analytics default plugins:

- `01-input-beats.conf`
- `01-input-syslog.conf`
- `01-input-snmp.conf`
- `01-input-http.conf`
- `01-input-file.conf`
- `01-input-database.conf`
- `020-filter-beats-syslog.conf`
- `020-filter-network.conf`
- `099-filter-geoip.conf`
- `100-output-elasticsearch.conf`
- `naemon_beat.example`
- `perflogs.example`

19.1 Logstash - Input "beats"

This plugin wait for receiving data from remote beats services. It use tcp /5044 port for communication:

```
        input {
                beats {
                        port => 5044
                }
        }
```

19.1.1 Getting data from share folder

Using beats, you can reading data from FTP, SFTP, SMB share. Connection to remote resources should be done as follows:

Input - FTP server

- Installation

```
yum install curlftpfs
```

- Create mount ftp directory

```
mkdir /mnt/my_ftp
```

- Use `curlftpfs` to mount your remote ftp site. Suppose my access credentials are as follows:

```
urlftpfs ftp-user:ftp-pass@my-ftp-location.local /mnt/my_ftp/
```

Input - SFTP server

- Install the required packages

```
yum install sshfs
```

- Add user

```
sudo adduser yourusername fuse
```

- Create local folder

```
mkdir ~/Desktop/sftp
```

- Mount remote folder to local:

```
sshfs HOSTuser@remote.host.or.ip:/host/dir/to/mount ~/Desktop/sftp
```

Input - SMB/CIFS server

- Create local folder

```
mkdir ~/Desktop/smb
```

- Mount remote folder to local:

```
mount -t smbfs //remoate.host.or.ip/freigabe /mnt -o username=testuser
```

or mount -t cifs //remoate.host.or.ip/freigabe /mnt -o username=testuser

19.2 Logstash - Input "network"

This plugin read events over a TCP or UDP socket assigns the appropriate tags:

```
input {
        tcp {
                port => 5514
                type => "network"

                tags => [ "LAN", "TCP" ]
        }

        udp {
                port => 5514
                type => "network"

                tags => [ "LAN", "UDP" ]
        }
}
```

19.3 Logstash - Input SNMP

The SNMP input polls network devices using Simple Network Management Protocol (SNMP) to gather information related to the current state of the devices operation:

```
input {
  snmp {
    get => ["1.3.6.1.2.1.1.1.0"]
    hosts => [{host => "udp:127.0.0.1/161" community => "public" version =>
↪"2c"  retries => 2  timeout => 1000}]
    }
}
```

19.4 Logstash - Input HTTP / HTTPS

Using this input you can receive single or multiline events over http(s). Applications can send an HTTP request to the endpoint started by this input and Logstash will convert it into an event for subsequent processing. Sample definition:

```
input {
 http {
    host => "0.0.0.0"
    port => "8080"
  }
}
```

Events are by default sent in plain text. You can enable encryption by setting ssl to true and configuring the ssl_certificate and ssl_key options:

```
        input {
         http {
            host => "0.0.0.0"
            port => "8080"
            ssl => "true"
            ssl_certificate => "path_to_certificate_file"
            ssl_key => "path_to_key_file"
          }
        }
```

19.5 Logstash - Input File

This plugin stream events from files, normally by tailing them in a manner similar to tail -0F but optionally reading them from the beginning. Sample definition:

```
        file {
            path => "/tmp/access_log"
            start_position => "beginning"
          }
```

19.6 Logstash - Input database

This plugin can read data in any database with a JDBC interface into Logstash. You can periodically schedule ingestion using a cron syntax (see schedule setting) or run the query one time to load data into Logstash. Each row in the resultset becomes a single event. Columns in the resultset are converted into fields in the event.

19.6.1 Logasth input - MySQL

Download jdbc driver: https://dev.mysql.com/downloads/connector/j/

Sample definition:

```
input {
  jdbc {
    jdbc_driver_library => "mysql-connector-java-5.1.36-bin.jar"
    jdbc_driver_class => "com.mysql.jdbc.Driver"
    jdbc_connection_string => "jdbc:mysql://localhost:3306/mydb"
    jdbc_user => "mysql"
    jdbc_password => "mysql"
    parameters => { "favorite_artist" => "Beethoven" }
    schedule => "* * * * *"
    statement => "SELECT * from songs where artist = :favorite_artist"
  }
}
```

19.6.2 Logasth input - MSSQL

Download jdbc driver: https://docs.microsoft.com/en-us/sql/connect/jdbc/download-microsoft-jdbc-driver-for-sql-server?view=sql-server-ver15

Sample definition:

```
input {
  jdbc {
    jdbc_driver_library => "./mssql-jdbc-6.2.2.jre8.jar"
    jdbc_driver_class => "com.microsoft.sqlserver.jdbc.SQLServerDriver"
    jdbc_connection_string => "jdbc:sqlserver://VB201001000;databaseName=Database;"
    jdbc_user => "mssql"
    jdbc_password => "mssql"
    jdbc_default_timezone => "UTC"
    statement_filepath => "/usr/share/logstash/plugin/query"
    schedule => "*/5 * * * *"
    sql_log_level => "warn"
    record_last_run => "false"
    clean_run => "true"
  }
}
```

19.6.3 Logstash input - Oracle

Download jdbc driver: https://www.oracle.com/database/technologies/appdev/jdbc-downloads.html

Sample definition:

```
input {
  jdbc {
    jdbc_driver_library => "./ojdbc8.jar"
    jdbc_driver_class => "oracle.jdbc.driver.OracleDriver"
    jdbc_connection_string => "jdbc:oracle:thin:@hostname:PORT/SERVICE"
    jdbc_user => "oracle"
    jdbc_password => "oracle"
    parameters => { "favorite_artist" => "Beethoven" }
    schedule => "* * * * *"
    statement => "SELECT * from songs where artist = :favorite_artist"
  }
}
```

19.6.4 Logstash input - PostgreSQL

Download jdbc driver: https://jdbc.postgresql.org/download.html

Sample definition:

```
input {
    jdbc {
        jdbc_driver_library => "D:/postgresql-42.2.5.jar"
        jdbc_driver_class => "org.postgresql.Driver"
        jdbc_connection_string => "jdbc:postgresql://127.0.0.1:57610/mydb"
        jdbc_user => "myuser"
        jdbc_password => "mypw"
        statement => "select * from mytable"
    }
}
```

19.7 Logstash - Filter "beats syslog"

This filter processing an event data with syslog type:

```
        filter {

        if [type] == "syslog" {
                grok {
                                match => {
                                  "message" => [
                                  # auth: ssh|sudo|su

                                  "%
→{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]}␣
→sshd(?:\[%{POSINT:[system][auth][pid]}\])?: %{DATA:[system][auth][ssh][event]} %
→{DATA:[system][auth][ssh][method]} for (invalid user )?%{DATA:[system][auth][user]}␣
→from %{IPORHOST:[system][auth][ssh][ip]} port %{NUMBER:[system][auth][ssh][port]}␣
→ssh2(: %{GREEDYDATA:[system][auth][ssh][signature]})?",

                                  "%
→{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]}␣
→sshd(?:\[%{POSINT:[system][auth][pid]}\])?: %{DATA:[system][auth][ssh][event]} user␣
→%{DATA:[system][auth][user]} from %{IPORHOST:[system][auth][ssh][ip]}",

                                  "%
→{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]}␣
→sshd(?:\[%{POSINT:[system][auth][pid]}\])?: Did not receive identification string␣
→from %{IPORHOST:[system][auth][ssh][dropped_ip]}",

                                  "%
→{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]}␣
→sudo(?:\[%{POSINT:[system][auth][pid]}\])?: \s*%{DATA:[system][auth][user]} :( %
→{DATA:[system][auth][sudo][error]} ;)? TTY=%{DATA:[system][auth][sudo][tty]} ; PWD=%
→{DATA:[system][auth][sudo][pwd]} ; USER=%{DATA:[system][auth][sudo][user]} ;␣
→COMMAND=%{GREEDYDATA:[system][auth][sudo][command]}",

                                  "%
→{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]} %
→{DATA:[system][auth][program]}(?:\[%{POSINT:[system][auth][pid]}\])?: %
→{GREEDYMULTILINE:[system][auth][message]}",

                                  # add/remove user or group
                                  "%
→{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]}␣
→groupadd(?:\[%{POSINT:[system][auth][pid]}\])?: new group: name=%{DATA:system.auth.
→groupadd.name}, GID=%{NUMBER:system.auth.groupadd.gid}",

                                  "%
→{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]}␣
→userdel(?:\[%{POSINT:[system][auth][pid]}\])?: removed group '%
→{DATA:[system][auth][groupdel][name]}' owned by '%
→{DATA:[system][auth][group][owner]}'",

                                  "%
→{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]}␣
→useradd(?:\[%{POSINT:[system][auth][pid]}\])?: new user: name=%
→{DATA:[system][auth][user][add][name]}, UID=%{NUMBER:[system][auth][user][add][uid]}
→, GID=%{NUMBER:[system][auth][user][add][gid]}, home=%
→{DATA:[system][auth][user][add][home]}, shell=%
→{DATA:[system][auth][user][add][shell]}$",
```

```
                                                          "%
→{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]}␣
→userdel(?:\[%{POSINT:[system][auth][pid]}\])?: delete user '%
→{WORD:[system][auth][user][del][name]}'$",

                                                          "%
→{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{SYSLOGHOST:[system][auth][hostname]}␣
→usermod(?:\[%{POSINT:[system][auth][pid]}\])?: add '%
→{WORD:[system][auth][user][name]}' to group '%{WORD:[system][auth][user][memberof]}'
→",

                                                          # yum install/erase/update package
                                                          "%
→{SYSLOGTIMESTAMP:[system][auth][timestamp]} %{DATA:[system][package][action]}: %
→{NOTSPACE:[system][package][name]}"
                                                        ]

                                            }

                                        pattern_definitions => {
                                            "GREEDYMULTILINE"=> "(.|\n)*"
                                        }
                                    }

                                    date {
                                                match => [ "[system][auth][timestamp]
→",

                                                "MMM  d HH:mm:ss",
                                            "MMM dd HH:mm:ss"
                                            ]
                                            target => "[system][auth][timestamp]"
                                    }

                                    mutate {
                                        convert => { "[system][auth][pid]" => "integer" }
                                        convert => { "[system][auth][groupadd][gid]" =>
→"integer" }

                                        convert => { "[system][auth][user][add][uid]" =>
→"integer" }

                                        convert => { "[system][auth][user][add][gid]" =>
→"integer" }
                                    }
                            }
                        }
```

19.8 Logstash - Filter "network"

This filter processing an event data with network type:

```
filter {
 if [type] == "network" {
    grok {
            named_captures_only => true
```

```
        match => {
            "message" => [

            # Cisco Firewall
            "%{SYSLOG5424PRI}%{NUMBER:log_sequence#}:%{SPACE}%
→{IPORHOST:device_ip}: (?:.)?%{CISCOTIMESTAMP:log_data} CET: %%{CISCO_
→REASON:facility}-%{INT:severity_level}-%{CISCO_REASON:facility_mnemonic}:%{SPACE}%
→{GREEDYDATA:event_message}",

            # Cisco Routers
            "%{SYSLOG5424PRI}%{NUMBER:log_sequence#}:%{SPACE}%
→{IPORHOST:device_ip}: (?:.)?%{CISCOTIMESTAMP:log_data} CET: %%{CISCO_
→REASON:facility}-%{INT:severity_level}-%{CISCO_REASON:facility_mnemonic}:%{SPACE}%
→{GREEDYDATA:event_message}",

            # Cisco Switches
            "%{SYSLOG5424PRI}%{NUMBER:log_sequence#}:%{SPACE}%
→{IPORHOST:device_ip}: (?:.)?%{CISCOTIMESTAMP:log_data} CET: %%{CISCO_
→REASON:facility}-%{INT:severity_level}-%{CISCO_REASON:facility_mnemonic}:%{SPACE}%
→{GREEDYDATA:event_message}",
            "%{SYSLOG5424PRI}%{NUMBER:log_sequence#}:%{SPACE}(?:.)?%
→{CISCOTIMESTAMP:log_data} CET: %%{CISCO_REASON:facility}-%{INT:severity_level}-%
→{CISCO_REASON:facility_mnemonic}:%{SPACE}%{GREEDYDATA:event_message}",

            # HP switches
            "%{SYSLOG5424PRI}%{SPACE}%{CISCOTIMESTAMP:log_data} %
→{IPORHOST:device_ip} %{CISCO_REASON:facility}:%{SPACE}%{GREEDYDATA:event_message}"
            ]

        }
    }

    syslog_pri { }

    if [severity_level] {

      translate {
        dictionary_path => "/etc/logstash/dictionaries/cisco_syslog_severity.yml"
        field => "severity_level"
        destination => "severity_level_descr"
      }

    }

    if [facility] {

      translate {
        dictionary_path => "/etc/logstash/dictionaries/cisco_syslog_facility.yml"
        field => "facility"
        destination => "facility_full_descr"
      }

    }

     #ACL
     if [event_message] =~ /(\d+.\d+.\d+.\d+)/ {
      grok {
```

```
        match => {
            "event_message" => [
                    "list %{NOTSPACE:[acl][name]} %{WORD:[acl][action]} %
→{WORD:[acl][proto]} %{IP:[src][ip]}.*%{IP:[dst][ip]}",
                    "list %{NOTSPACE:[acl][name]} %{WORD:[acl][action]} %
→{IP:[src][ip]}",
                    "^list %{NOTSPACE:[acl][name]} %{WORD:[acl][action]} %
→{WORD:[acl][proto]} %{IP:[src][ip]}.*%{IP:[dst][ip]}"
                    ]
        }
    }
}

    if [src][ip] {

            cidr {
                address => [ "%{[src][ip]}" ]
                network => [ "0.0.0.0/32", "10.0.0.0/8", "172.16.0.0/12", "192.168.
→0.0/16", "fc00::/7", "127.0.0.0/8", "::1/128", "169.254.0.0/16", "fe80::/10","224.0.
→0.0/4", "ff00::/8","255.255.255.255/32"  ]
                add_field => { "[src][locality]" => "private" }
            }

            if ![src][locality] {
                mutate {
                    add_field => { "[src][locality]" => "public" }
                }
            }
    }


    if [dst][ip] {
            cidr {
                address => [ "%{[dst][ip]}" ]
                network => [ "0.0.0.0/32", "10.0.0.0/8", "172.16.0.0/12", "192.168.
→0.0/16", "fc00::/7", "127.0.0.0/8", "::1/128",
                            "169.254.0.0/16", "fe80::/10","224.0.0.0/4", "ff00::/8
→","255.255.255.255/32" ]
                add_field => { "[dst][locality]" => "private" }
            }

            if ![dst][locality] {
                mutate {
                    add_field => { "[dst][locality]" => "public" }
                }
            }
    }

    # date format
    date {
      match => [ "log_data",
            "MMM dd HH:mm:ss",
            "MMM  dd HH:mm:ss",
            "MMM dd HH:mm:ss.SSS",
            "MMM  dd HH:mm:ss.SSS",
            "ISO8601"
        ]
```

```
            target => "log_data"
        }

 }
}
```

19.9 Logstash - Filter "geoip"

This filter processing an events data with IP address and check localization:

```
filter {
    if [src][locality] == "public" {

            geoip {
                    source => "[src][ip]"
                    target => "[src][geoip]"
                    database => "/etc/logstash/geoipdb/GeoLite2-City.mmdb"
                    fields => [ "city_name", "country_name", "continent_code",
→"country_code2", "location" ]
                    remove_field => [ "[src][geoip][ip]" ]
            }

            geoip {
                    source => "[src][ip]"
                    target => "[src][geoip]"
                    database => "/etc/logstash/geoipdb/GeoLite2-ASN.mmdb"
                    remove_field => [ "[src][geoip][ip]" ]
            }

    }

    if [dst][locality] == "public" {

            geoip {
                    source => "[dst][ip]"
                    target => "[dst][geoip]"
                    database => "/etc/logstash/geoipdb/GeoLite2-City.mmdb"
                    fields => [ "city_name", "country_name", "continent_code",
→"country_code2", "location" ]
                    remove_field =>  [ "[dst][geoip][ip]" ]
            }

            geoip {
                    source => "[dst][ip]"
                    target => "[dst][geoip]"
                    database => "/etc/logstash/geoipdb/GeoLite2-ASN.mmdb"
                    remove_field => [ "[dst][geoip][ip]" ]
            }
    }

}
```

19.10 Logstash - Output to Elasticsearch

This output plugin sends all data to the local Elasticsearch instance and create indexes:

```
output {
        elasticsearch {
            hosts => [ "127.0.0.1:9200" ]

            index => "%{type}-%{+YYYY.MM.dd}"

            user => "logstash"
            password => "logstash"
        }
}
```

19.11 Logstash plugin for "naemon beat"

This Logstash plugin has example of complete configuration for integration with *naemon* application:

```
input {
    beats {
        port => FILEBEAT_PORT
        type => "naemon"
    }
}

filter {
    if [type] == "naemon" {
        grok {
            patterns_dir => [ "/etc/logstash/patterns" ]
            match => { "message" => "%{NAEMONLOGLINE}" }
            remove_field => [ "message" ]
        }
        date {
            match => [ "naemon_epoch", "UNIX" ]
            target => "@timestamp"
            remove_field => [ "naemon_epoch" ]
        }
    }
}

output {
    # Single index
#    if [type] == "naemon" {
#        elasticsearch {
#            hosts => ["ELASTICSEARCH_HOST:ES_PORT"]
#            index => "naemon-%{+YYYY.MM.dd}"
#        }
#    }

    # Separate indexes
    if [type] == "naemon" {
        if "_grokparsefailure" in [tags] {
            elasticsearch {
                hosts => ["ELASTICSEARCH_HOST:ES_PORT"]
```

(continues on next page)

```
                    index => "naemongrokfailure"
            }
        }
        else {
            elasticsearch {
                hosts => ["ELASTICSEARCH_HOST:ES_PORT"]
                index => "naemon-%{+YYYY.MM.dd}"
            }
        }
    }
}
```

19.12 Logstash plugin for "perflog"

This Logstash plugin has example of complete configuration for integration with perflog:

```
input {
  tcp {
    port => 6868
    host => "0.0.0.0"
    type => "perflogs"
  }
}

filter {
  if [type] == "perflogs" {
    grok {
      break_on_match => "true"
      match => {
        "message" => [
          "DATATYPE::%{WORD:datatype}\tTIMET::%{NUMBER:timestamp}\tHOSTNAME::%
→{DATA:hostname}\tSERVICEDESC::%{DATA:servicedescription}\tSERVICEPERFDATA::%
→{DATA:performance}\tSERVICECHECKCOMMAND::.*?HOSTSTATE::%{WORD:hoststate}
→\tHOSTSTATETYPE::.*?SERVICESTATE::%{WORD:servicestate}\tSERVICESTATETYPE::%
→{WORD:servicestatetype}",
          "DATATYPE::%{WORD:datatype}\tTIMET::%{NUMBER:timestamp}\tHOSTNAME::%
→{DATA:hostname}\tHOSTPERFDATA::%{DATA:performance}\tHOSTCHECKCOMMAND::.*?HOSTSTATE::
→%{WORD:hoststate}\tHOSTSTATETYPE::%{WORD:hoststatetype}"
        ]
      }
      remove_field => [ "message" ]
    }
    kv {
      source => "performance"
      field_split => "\t"
      remove_char_key => "\.\'"
      trim_key => " "
      target => "perf_data"
      remove_field => [ "performance" ]
      allow_duplicate_values => "false"
      transform_key => "lowercase"
    }
    date {
      match => [ "timestamp", "UNIX" ]
```

```
      target => "@timestamp"
      remove_field => [ "timestamp" ]
    }
  }
}

output {
  if [type] == "perflogs" {
    elasticsearch {
      hosts => ["127.0.0.1:9200"]
      index => "perflogs-%{+YYYY.MM.dd}"
    }
  }
}
```

19.13 Single password in all Logstash outputs

You can set passwords and other Logstash pipeline settings as environment variables. This can be useful if the password was changed for the `logastash` user and it must be to update in the configuration files.

Configuration steps:

1. Create the service file:

   mkdir –p /etc/systemd/system/logstash.service.d vi /etc/systemd/system/logstash.service.d/logstash.conf

   ```
           [Service]
           Environment="ELASTICSEARCH_ES_USER=logserver"
           Environment="ELASTICSEARCH_ES_PASSWD=logserver"
   ```

2. Reload systemctl daemon:

   ```
   systemctl daemon-reload
   ```

3. Sample definition of Logstash output pipline seciotn:

   ```
   output   {
     elasticsearch {
       index => "test-%{+YYYY.MM.dd}"
       user => "${ELASTICSEARCH_ES_USER:elastic}"
       password => "${ELASTICSEARCH_ES_PASSWD:changeme}"
     }
   }
   ```

19.14 Secrets keystore for secure settings

When you configure Logstash, you can use the Logstash keystore to securely store secret values for use in configuration settings (passwords, usernames, other settings).

Configuration steps:

1. Set the keystore password

```
vi /etc/sysconfi/logstash
LOGSTASH_KEYSTORE_PASS=keystorepass
```

2. Create the new keystore:

```
/usr/share/logstash/bin/logstash-keystore create --path.settings /etc/logstash
```

During createation keystore you can provide the keysore passowrd

3. Add new entry to keystore:

```
usr/share/logstash/bin/logstash-keystore add ES_PWD --path.settings /etc/logstash
```

When adding an entry to the keystore, set the value of the entry.

4. Listing added entries:

```
/usr/share/logstash/bin/logstash-keystore list --path.settings /etc/logstash
```

5. Removing entries:

```
/usr/share/logstash/bin/logstash-keystore remove ES_PWD --path.settings /etc/
↪logstash
```
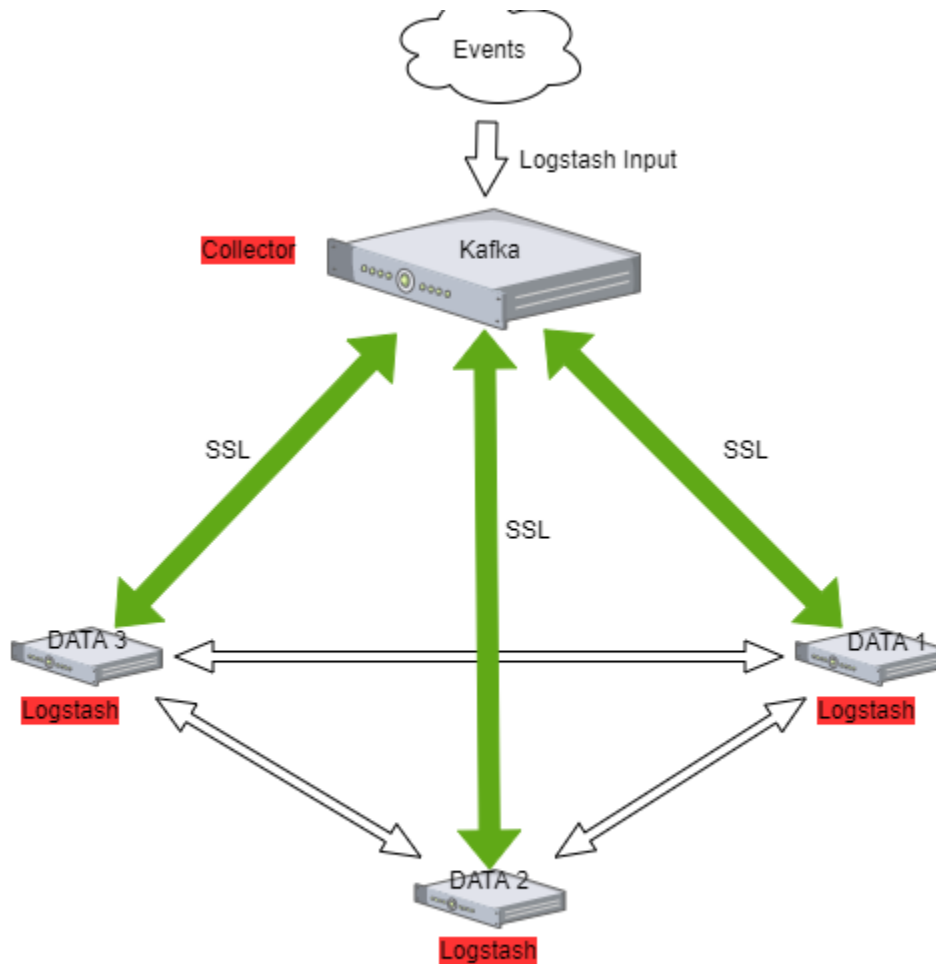
Sample definition of Logstash output pipline seciotn:

```
output {
  elasticsearch {
    index => "test-%{+YYYY.MM.dd}"
    user => "${ES_PWD}"
    password => "${ES_PWD}"
  }
}
```

19.15 Enabling encryption for Apache Kafka clients

Kafka allows you to distribute the load between nodes receiving data and encrypts communication.

Architecture example:

19.15.1 The Kafka installation

Documentation during creation.

19.15.2 Enabling encryption in Kafka

Generate SSL key and certificate for each Kafka broker

```
keytool -keystore server.keystore.jks -alias localhost -validity {validity} -genkey -
→keyalg RSA
```

Configuring Host Name In Certificates

```
keytool -keystore server.keystore.jks -alias localhost -validity {validity} -genkey -
→keyalg RSA -ext SAN=DNS:{FQDN}
```

Verify content of the generated certificate:

```
keytool -list -v -keystore server.keystore.jks
```

Creating your own CA

```
openssl req -new -x509 -keyout ca-key -out ca-cert -days 365
keytool -keystore client.truststore.jks -alias CARoot -import -file ca-cert
keytool -keystore server.truststore.jks -alias CARoot -import -file ca-cert
```

Signing the certificate

```
keytool -keystore server.keystore.jks -alias localhost -certreq -file cert-file
openssl x509 -req -CA ca-cert -CAkey ca-key -in cert-file -out cert-signed -days
→{validity} -CAcreateserial -passin pass:{ca-password}
```

Import both the certificate of the CA and the signed certificate into the keystore

```
keytool -keystore server.keystore.jks -alias CARoot -import -file ca-cert
keytool -keystore server.keystore.jks -alias localhost -import -file cert-signed
```

19.15.3 Configuring Kafka Brokers

In `/etc/kafka/server.properties` file set the following options:

```
listeners=PLAINTEXT://host.name:port,SSL://host.name:port

ssl.keystore.location=/var/private/ssl/server.keystore.jks
ssl.keystore.password=test1234
ssl.key.password=test1234
ssl.truststore.location=/var/private/ssl/server.truststore.jks
ssl.truststore.password=test1234
```

and restart the Kafka service

```
systemctl restart kafka
```

19.15.4 Configuring Kafka Clients

Logstash

Configure the output section in Logstash based on the following example:

```
output {
  kafka {
    bootstrap_servers => "host.name:port"
    security_protocol => "SSL"
    ssl_truststore_type => "JKS"
    ssl_truststore_location => "/var/private/ssl/client.truststore.jks"
    ssl_truststore_password => "test1234"
    client_id => "host.name"
    topic_id => "Topic-1"
    codec => json
  }
}
```

Configure the input section in Logstash based on the following example:

```
input {
  kafka {
    bootstrap_servers => "host.name:port"
    security_protocol => "SSL"
    ssl_truststore_type => "JKS"
    ssl_truststore_location => "/var/private/ssl/client.truststore.jks"
    ssl_truststore_password => "test1234"
    consumer_threads => 4
    topics => [ "Topic-1" ]
    codec => json
    tags => ["kafka"]
  }
}
```

Integrations

20.1 OP5 - Naemon logs

20.1.1 Logstash

1. In ITRS Log Analytics `naemon_beat.conf` set up `ELASTICSEARCH_HOST`, `ES_PORT`, `FILEBEAT_PORT`

2. Copy ITRS Log Analytics `naemon_beat.conf` to `/etc/logstash/conf.d`

3. Based on "FILEBEAT_PORT" if firewall is running:

```
sudo firewall-cmd --zone=public --permanent --add-port=FILEBEAT_PORT/tcp
sudo firewall-cmd --reload
```

1. Based on amount of data that elasticsearch will receive you can also choose whether you want index creation to be based on moths or days:

```
index => "ITRS Log Analytics-naemon-%{+YYYY.MM}"
or
index => "ITRS Log Analytics-naemon-%{+YYYY.MM.dd}"
```

1. Copy `naemon` file to `/etc/logstash/patterns` and make sure it is readable by logstash process

2. Restart *logstash* configuration e.g.:

```
sudo systemct restart logstash
```

20.1.2 Elasticsearch

1. Connect to Elasticsearch node via SSH and Install index pattern for naemon logs. Note that if you have a default pattern covering *settings* section you should delete/modify that in naemon_template.sh:

```
"settings": {
  "number_of_shards": 5,
  "auto_expand_replicas": "0-1"
},
```

1. Install template by running: `./naemon_template.sh`

20.1.3 ITRS Log Analytics Monitor

1. On ITRS Log Analytics Monitor host install filebeat (for instance via rpm `https://www.elastic.co/downloads/beats/filebeat`)

2. In `/etc/filebeat/filebeat.yml` add:

```
#========================== Filebeat inputs ===============================
filebeat.config.inputs:
  enabled: true
  path: configs/*.yml
```

3. You also will have to configure the output section in `filebeat.yml`. You should have one logstash output:

```
#-------------------------- Logstash output ------------------------------
output.logstash:
  # The Logstash hosts
  hosts: ["LOGSTASH_IP:FILEBEAT_PORT"]
```

If you have few logstash instances - `Logstash` section has to be repeated on every node and `hosts:` should point to all of them:

```
hosts: ["LOGSTASH_IP:FILEBEAT_PORT", "LOGSTASH_IP:FILEBEAT_PORT", "LOGSTASH_
↪IP:FILEBEAT_PORT" ]
```

4. Create `/etc/filebeat/configs` catalog.

5. Copy `naemon_logs.yml` to a newly created catalog.

6. Check the newly added configuration and connection to logstash. Location of executable might vary based on os:

```
/usr/share/filebeat/bin/filebeat --path.config /etc/filebeat/ test config
/usr/share/filebeat/bin/filebeat --path.config /etc/filebeat/ test output
```

7. Restart filebeat:

```
sudo systemctl restart filebeat # RHEL/CentOS 7
sudo service filebeat restart # RHEL/CentOS 6
```

20.1.4 Elasticsearch

At this moment there should be a new index on the Elasticsearch node:

```
curl -XGET '127.0.0.1:9200/_cat/indices?v'
```

Example output:

```
     health status index                      uuid                    pri rep docs.count␣
→docs.deleted store.size pri.store.size
     green  open    ITRS Log Analytics-naemon-2018.11    gO8XRsHiTNm63nI_RVCy8w   1␣
→   0      23176              0      8.3mb          8.3mb
```

If the index has been created, in order to browse and visualise the data, "index pattern" needs to be added in Kibana.

20.2 OP5 - Performance data

Below instruction requires that between ITRS Log Analytics node and Elasticsearch node is working Logstash instance.

20.2.1 Elasticsearch

1. First, settings section in *ITRS Log Analyticstemplate.sh* should be adjusted, either:

   - there is a default template present on Elasticsearch that already covers shards and replicas then settings sections should be removed from the *ITRS Log Analyticstemplate.sh* before executing

   - there is no default template - shards and replicas should be adjusted for you environment (keep in mind replicas can be added later, while changing shards count on existing index requires reindexing it)

   ```
   "settings": {
     "number_of_shards": 5,
     "number_of_replicas": 0
   }
   ```

2. In URL *ITRS Log Analyticsperfdata* is a name for the template - later it can be search for or modify with it.

3. The "*template*" is an index pattern. New indices matching it will have the settings and mapping applied automatically (change it if you index name for *ITRS Log Analytics perfdata* is different).

4. Mapping name should match documents type:

   ```
   "mappings": {
     "ITRS Log Analyticsperflogs"
   ```

5. Running ITRS Log Analyticstemplate.sh will create a template (not index) for ITRS Log Analytics perf data documents.

20.2.2 Logstash

1. The *ITRS Log Analyticsperflogs.conf* contains example of *input/filter/output* configuration. It has to be copied to */etc/logstash/conf.d/*. Make sure that the *logstash* has permissions to read the configuration files:

   ```
   chmod 664 /etc/logstash/conf.d/ITRS Log Analyticsperflogs.conf
   ```

2. In the input section comment/uncomment *"beats"* or *"tcp"* depending on preference (beats if *Filebeat* will be used and tcp if *NetCat*). The port and the type has to be adjusted as well:

   ```
   port => PORT_NUMBER
   type => "ITRS Log Analyticsperflogs"
   ```

3. In a filter section type has to be changed if needed to match the input section and Elasticsearch mapping.

4. In an output section type should match with the rest of a *config*. host should point to your elasticsearch node. index name should correspond with what has been set in elasticsearch template to allow mapping application. The date for index rotation in its name is recommended and depending on the amount of data expecting to be transferred should be set to daily (+YYYY.MM.dd) or monthly (+YYYY.MM) rotation:

```
hosts => ["127.0.0.1:9200"]
index => "ITRS Log Analytics-perflogs-%{+YYYY.MM.dd}"
```

5. Port has to be opened on a firewall:

```
sudo firewall-cmd --zone=public --permanent --add-port=PORT_NUMBER/tcp
sudo firewall-cmd --reload
```

6. Logstash has to be reloaded:

```
sudo systemctl restart logstash
```

or

```
sudo kill -1 LOGSTASH_PID
```

20.2.3 ITRS Log Analytics Monitor

1. You have to decide wether FileBeat or NetCat will be used. In case of Filebeat - skip to the second step. Otherwise:

   • Comment line:

```
  54    open(my $logFileHandler, '>>', $hostPerfLogs) or die "Could not open
→$hostPerfLogs"; #FileBeat
   •        Uncomment lines:
  55 #    open(my $logFileHandler, '>', $hostPerfLogs) or die "Could not open
→$hostPerfLogs"; #NetCat
  ...
  88 #    my $logstashIP = "LOGSTASH_IP";
  89 #    my $logstashPORT = "LOGSTASH_PORT";
  90 #    if (-e $hostPerfLogs) {
  91 #        my $pid1 = fork();
  92 #        if ($pid1 == 0) {
  93 #            exec("/bin/cat $hostPerfLogs | /usr/bin/nc -w 30
→$logstashIP $logstashPORT");
  94 #        }
  95 #    }
```

   • In process-service-perfdata-log.pl and process-host-perfdata-log.pl: change logstash IP and port:

```
  92 my $logstashIP = "LOGSTASH_IP";
  93 my $logstashPORT = "LOGSTASH_PORT";
```

2. In case of running single ITRS Log Analytics node, there is no problem with the setup. In case of a peered environment *$do_on_host* variable has to be set up and the script *process-service-perfdata-log.pl/process-host-perfdata-log.pl* has to be propagated on all of ITRS Log Analytics nodes:

```
 16 $do_on_host = "EXAMPLE_HOSTNAME"; # ITRS Log Analytics node name to run the
→script on
 17 $hostName = hostname; # will read hostname of a node running the script
```

3. Example of command definition (*/opt/monitor/etc/checkcommands.cfg*) if scripts have been copied to */opt/plugins/custom/*:

```
# command 'process-service-perfdata-log'
define command{
    command_name                      process-service-perfdata-log
    command_line                      /opt/plugins/custom/process-service-perfdata-
↪log.pl $TIMET$
    }
# command 'process-host-perfdata-log'
define command{
    command_name                      process-host-perfdata-log
    command_line                      /opt/plugins/custom/process-host-perfdata-log.
↪pl $TIMET$
    }
```

4. In */opt/monitor/etc/naemon.cfg* *service_perfdata_file_processing_command* and *host_perfdata_file_processing_command* has to be changed to run those custom scripts:

```
service_perfdata_file_processing_command=process-service-perfdata-log
host_perfdata_file_processing_command=process-host-perfdata-log
```

5. In addition *service_perfdata_file_template* and *host_perfdata_file_template* can be changed to support sending more data to Elasticsearch. For instance, by adding *$HOSTGROUPNAMES$* and *$SERVICEGROUPNAMES$* macros logs can be separated better (it requires changes to Logstash filter config as well)

6. Restart naemon service:

```
sudo systemctl restart naemon # CentOS/RHEL 7.x
sudo service naemon restart # CentOS/RHEL 7.x
```

7. If *FileBeat* has been chosen, append below to *filebeat.conf* (adjust IP and PORT):

```
filebeat.inputs:
- type: log
  enabled: true
  paths:
    - /opt/monitor/var/service_performance.log
    - /opt/monitor/var/host_performance.log

        tags: ["ITRS Log Analyticsperflogs"]


output.logstash:
  # The Logstash hosts
  hosts: ["LOGSTASH_IP:LOGSTASH_PORT"]
```

- Restart FileBeat service:

```
sudo systemctl restart filebeat # CentOS/RHEL 7.x
sudo service filebeat restart # CentOS/RHEL 7.x
```

20.2.4 Kibana

At this moment there should be new index on the Elasticsearch node with performance data documents from ITRS Log Analytics Monitor. Login to an Elasticsearch node and run: `curl -XGET '127.0.0.1:9200/_cat/indices?v'` Example output:

```
health status index                           pri rep docs.count docs.deleted store.size␣
→pri.store.size
green  open   auth                            5   0          7          6230      1.8mb   ␣
→    1.8mb
green  open   ITRS Log Analytics-perflogs-2018.09.14   5   0      72109               0␣
→    24.7mb          24.7mb
```

After a while, if there is no new index make sure that:

- Naemon is runnig on ITRS Log Analytics node

- Logstash service is running and there are no errors in: */var/log/logstash/logstash-plain.log*

- Elasticsearch service is running an there are no errors in: */var/log/elasticsearch/elasticsearch.log*

If the index has been created, in order to browse and visualize the data "*index pattern*" needs to be added to Kibana.

1. After logging in to Kibana GUI go to *Settings* tab and add *ITRS Log Analytics-perflogs-** pattern. Chose *@timestamp* time field and click *Create*.

2. Performance data logs should be now accessible from Kibana GUI Discovery tab ready to be visualize.

20.3 OP5 Beat

The op5beat is small agent for collecting metrics from op5 Monitor.

The op5beat is located in the installation directory: `utils/op5integration/op5beat`

20.3.1 Installation for Centos7 and newer

1. Copy the necessary files to the appropriate directories:

```
cp -rf etc/* /etc/
cp -rf usr/* /usr/
cp -rf var/* /var/
```

1. Configure and start op5beat service (systemd):

```
cp -rf op5beat.service /usr/lib/systemd/system/
systemctl daemon-reload
systemctl enable op5beat
systemctl start op5beat
```

20.3.2 Installation for Centos6 and older

1. Copy the necessary files to the appropriate directories:

```
cp -rf etc/* /etc/
cp -rf usr/* /usr/
cp -rf var/* /var/
```

1. Configure and start op5beat service:

- sysV init:

```
cp -rf op5beat.service /etc/rc.d/init.d/op5beat
chkconfig op5beat on
service op5beat start
```

- supervisord (optional):

```
yum install supervisor
cp -rf supervisord.conf /etc/supervisord.conf
```

20.4 The Grafana instalation

1. To install the Grafana application you should:

   - add necessary repository to operating system:

```
[root@logserver-6 ~]# cat /etc/yum.repos.d/grafan.repo
[grafana]
name=grafana
baseurl=https://packagecloud.io/grafana/stable/el/7/$basearch
repo_gpgcheck=1
enabled=1
gpgcheck=1
gpgkey=https://packagecloud.io/gpg.key https://grafanarel.s3.amazonaws.com/
↪RPM-GPG-KEY-grafana
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
[root@logserver-6 ~]#
```

   - install the Grafana with following commands:

```
[root@logserver-6 ~]# yum search grafana
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: ftp.man.szczecin.pl
 * extras: centos.slaskdatacenter.com
 * updates: centos.slaskdatacenter.com
␣
↪================================================================================================
↪N/S matched: grafana␣
↪================================================================================================
 grafana.x86_64 : Grafana
 pcp-webapp-grafana.noarch : Grafana web application for Performance Co-
↪Pilot (PCP)

   Name and summary matches only, use "search all" for everything.

[root@logserver-6 ~]# yum install grafana
```

   - to run application use following commands:

```
[root@logserver-6 ~]# systemctl enable grafana-server
Created symlink from /etc/systemd/system/multi-user.target.wants/grafana-
↪server.service to /usr/lib/systemd/system/grafana-server.service.
[root@logserver-6 ~]#
[root@logserver-6 ~]# systemctl start grafana-server
```

(continues on next page)

```
  [root@logserver-6 ~]# systemctl status grafana-server
   grafana-server.service - Grafana instance
     Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; enabled;
→vendor preset: disabled)
     Active: active (running) since Thu 2018-10-18 10:41:48 CEST; 5s ago
       Docs: http://docs.grafana.org
  Main PID: 1757 (grafana-server)
     CGroup: /system.slice/grafana-server.service
             └─1757 /usr/sbin/grafana-server --config=/etc/grafana/grafana.
→ini --pidfile=/var/run/grafana/grafana-server.pid cfg:default.paths.logs=/
→var/log/grafana cfg:default.paths.data=/var/lib/grafana cfg:default.paths.
→plugins=/var...

   [root@logserver-6 ~]#
```

2. To connect the Grafana application you should:

   - define the default login/password (line 151;154 in config file)

```
  [root@logserver-6 ~]# cat /etc/grafana/grafana.ini
  .....
  148 ################################## Security ##########################
→##########
  149 [security]
  150 # default admin user, created on startup
  151 admin_user = admin
  152
  153 # default admin password, can be changed before first start of grafana,
→ or in profile settings
  154 admin_password = admin
  155
```
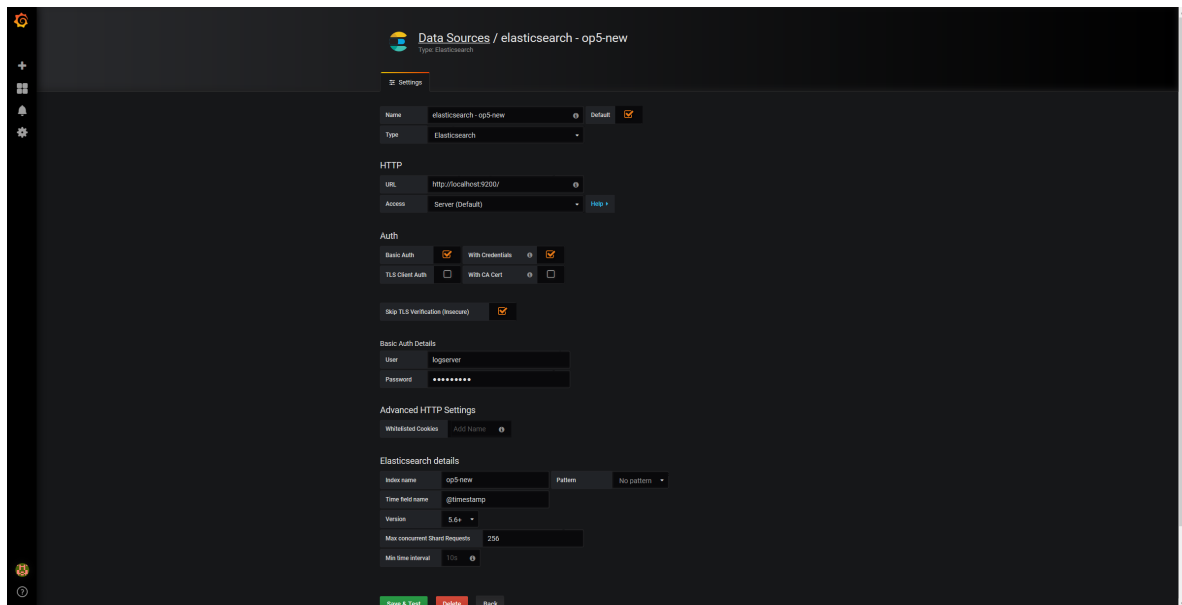
….. - restart *grafana-server* service:

```
              [root@logserver-6 ~]# systemctl restart grafana-server
- Login to Grafana user interface using web browser: *http://ip:3000*
```
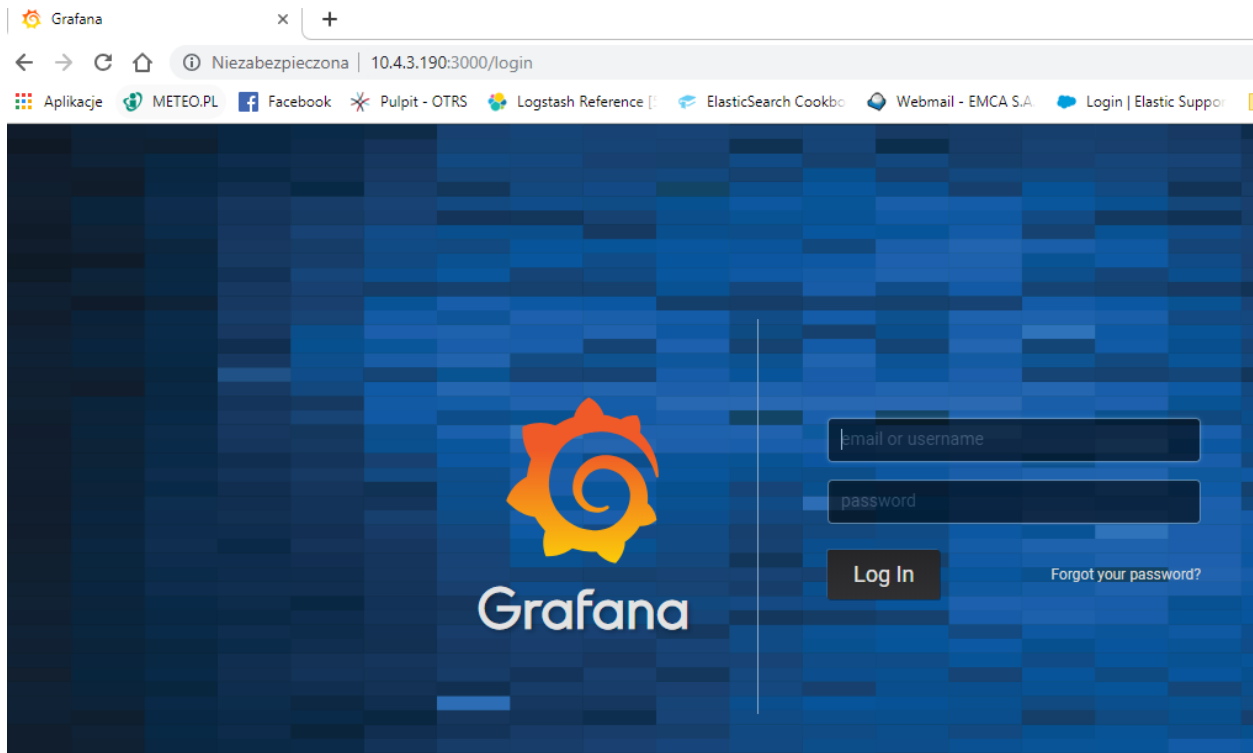
```
- use login and password that you set in the config file.
```

1. Use below example to set conection to Elasticsearch server:

20.5 The Beats configuration

20.5.1 Kibana API

Reference link: https://www.elastic.co/guide/en/kibana/master/api.html

After installing any of beats package you can use ready to use dashboard related to this beat package. For instance dashboard and index pattern are available in */usr/share/filebeat/kibana/6/* directory on Linux.

Before uploading index-pattern or dashboard you have to authorize yourself:

1. Set up *login/password/kibana_ip* variables, e.g.:

```
login=logserver
password=my_password
kibana_ip=10.4.11.243
```

2. Execute command which will save authorization cookie:

```
curl -c authorization.txt -XPOST -k "https://${kibana_ip}:5601/login" -d
→"username=${username}&password=${password}&version=6.2.3&location=https%3A%2F%2F
→${kibana_ip}%3A5601%2Flogin"
```

3. Upload index-pattern and dashboard to *Kibana*, e.g.:

```
curl -b authorization.txt -XPOST -k "https://${kibana_ip}:5601/api/kibana/
→dashboards/import" -H 'kbn-xsrf: true' -H 'Content-Type: application/json' -d@/
→usr/share/filebeat/kibana/6/index-pattern/filebeat.json
curl -b authorization.txt -XPOST -k "https://${kibana_ip}:5601/api/kibana/
→dashboards/import" -H 'kbn-xsrf: true' -H 'Content-Type: application/json' -d@/
→usr/share/filebeat/kibana/6/dashboard/Filebeat-mysql.json
```

4. When you want to upload beats index template to Ealsticsearch you have to recover it first (usually you do not send logs directly to Es rather than to Logstash first):

```
/usr/bin/filebeat export template --es.version 6.2.3 >> /path/to/beats_template.
→json
```

5. After that you can upload it as any other template (Access Es node with SSH):

```
curl -XPUT "localhost:9200/_template/ITRS Log Analyticsperfdata" -H'Content-Type:
→application/json' -d@beats_template.json
```

20.6 Wazuh integration

ITRS Log Analytics can integrate with the Wazuh, which is lightweight agent is designed to perform a number of tasks with the objective of detecting threats and, when necessary, trigger automatic responses. The agent core capabilities are:

- Log and events data collection
- File and registry keys integrity monitoring
- Inventory of running processes and installed applications
- Monitoring of open ports and network configuration

- Detection of rootkits or malware artifacts
- Configuration assessment and policy monitoring
- Execution of active responses

The Wazuh agents run on many different platforms, including Windows, Linux, Mac OS X, AIX, Solaris and HP-UX. They can be configured and managed from the Wazuh server.

20.6.1 Deploying Wazuh Server

https://documentation.wazuh.com/current/installation-guide/installing-wazuh-server/index.html#

20.6.2 Deploing Wazuh Agent

https://documentation.wazuh.com/current/installation-guide/installing-wazuh-agent/index.html

20.6.3 Filebeat configuration

20.7 BRO integration

20.8 2FA authorization with Google Auth Provider (example)

20.8.1 Software used (tested versions):

- NGiNX (1.16.1 - from CentOS base reposiory)
- oauth2_proxy (https://github.com/pusher/oauth2_proxy/releases - 4.0.0)

20.8.2 The NGiNX configuration:

1. Copy the ng_oauth2_proxy.conf to /etc/nginx/conf.d/;
2. Set ssl_certificate and ssl_certificate_key path in ng_oauth2_proxy.conf

When SSL is set using nginx proxy, Kibana can be started with http. However, if it is to be run with encryption, you also need to change proxy_pass to the appropriate one.

20.8.3 The oauth2_proxy configuration:

1. Create a directory in which the program will be located and its configuration:

```bash
mkdir -p /usr/share/oauth2_proxy/
mkdir -p /etc/oauth2_proxy/
```

2. Copy files to directories:

```bash
cp oauth2_proxy /usr/share/oauth2_proxy/
cp oauth2_proxy.cfg /etc/oauth2_proxy/
```

3. Set directives according to OAuth configuration in Google Cloud project

```
cfg
client_id =
client_secret =
# the following limits domains for authorization (* - all)
email_domains = [
  "*"
]
```

4. Set the following according to the public hostname:

cookie_domain = "kibana-host.org"

1. In case og-in restrictions for a specific group defined on the Google side:

   • Create administrative account: https://developers.google.com/identity/protocols/OAuth2ServiceAccount ;

   • Get configuration to JSON file and copy Client ID;

   • On the dashboard of the Google Cloud select "APIs & Auth" -> "APIs";

   • Click on "Admin SDK" and "Enable API";

   • Follow the instruction at https://developers.google.com/admin-sdk/directory/v1/guides/delegation#delegate_domain-wide_authority_to_your_service_account and give the service account the following permissions:

     ```
     https://www.googleapis.com/auth/admin.directory.group.readonly
     https://www.googleapis.com/auth/admin.directory.user.readonly
     ```

   • Follow the instructions to grant access to the Admin API https://support.google.com/a/answer/60757

   • Create or select an existing administrative email in the Gmail domain to flag it `google-admin-email`

   • Create or select an existing group to flag it `google-group`

   • Copy the previously downloaded JSON file to `/etc/oauth2_proxy/`.

   • In file oauth2_proxy set the appropriate path:

     ```
     google_service_account_json =
     ```

20.8.4 Service start up

• Start the NGiNX service

• Start the oauth2_proxy service

```
bash
/usr/share/oauth2_proxy/oauth2_proxy -config="/etc/oauth2_proxy/oauth2_proxy.cfg
↪"
```

In the browser enter the address pointing to the server with the Logserver installation

20.9 Cerebro - Elasticsearch web admin tool

20.9.1 Software Requirements

1. Cerebro v0.8.4

```bash
bash
wget 'https://github.com/lmenezes/cerebro/releases/download/v0.8.4/cerebro-0.8.4.
↪tgz'
```

2. Java 11+ [for basic-auth setup]

```bash
bash
yum install java-11-openjdk-headless.x86_64
```

3. Java 1.8.0 [without authorization]

```bash
bash
yum install java-1.8.0-openjdk-headless
```

20.9.2 Firewall Configuration

```bash
bash
firewall-cmd --permanent --add-port=5602/tcp
firewall-cmd --reload
```

20.9.3 Cerebro Configuration

1. Extract archive & move directory

```bash
bash
tar -xvf cerebro-0.8.4.tgz -C /opt/
mv /opt/cerebro-0.8.4/ /opt/cerebro
```

2. Add Cerebro service user

```bash
bash
useradd -M -d /opt/cerebro -s /sbin/nologin cerebro
```

3. Change Cerbero permissions

```bash
bash
chown -R cerebro:cerebro /opt/cerebro && chmod -R 700 /opt/cerebro
```

4. Install Cerbero service (cerebro.service):

```
[Unit]
Description=Cerebro

[Service]
Type=simple
User=cerebro
Group=cerebro
ExecStart=/opt/cerebro/bin/cerebro "-Dconfig.file=/opt/cerebro/conf/application.
↪conf"
Restart=always
WorkingDirectory=/opt/cerebro

[Install]
```

(continues on next page)

```
WantedBy=multi-user.target


bash
cp cerebro.service /usr/lib/systemd/system/
systemctl daemon-reload
systemctl enable cerebro
```

5. Customize configuration file: /opt/cerebro/conf/application.conf

   - Authentication

```
auth = {
  type: basic
    settings: {
      username = "logserver"
      password = "logserver"
    }
}
```

   - A list of known Elasticsearch hosts

```
hosts = [
  {
    host = "http://localhost:9200"
    name = "logserver"
    auth = {
      username = "username"
      password = "password"
    }
  }
]
```

   If needed uses secure connection (SSL) with Elasticsearch, set the following section that contains path to certificate. And change the host definition from `http` to `https`:

```
        play.ws.ssl {
          trustManager = {
            stores = [
              { type = "PEM", path = "/etc/elasticsearch/ssl/rootCA.crt" }
            ]
          }
        }
        play.ws.ssl.loose.acceptAnyCertificate=true
```

   - SSL access to cerebro

```
http = {
  port = "disabled"
}
https = {
  port = "5602"
}

# SSL access to cerebro - no self signed certificates
#play.server.https {
#   keyStore = {
```

```
#    path = "keystore.jks",
#    password = "SuperSecretKeystorePassword"
#  }
#}

#play.ws.ssl {
#  trustManager = {
#    stores = [
#      { type = "JKS", path = "truststore.jks", password =
→"SuperSecretTruststorePassword"  }
#    ]
#  }
#}
```

6. Start the service

```
bash
systemctl start cerebro
goto: https://127.0.0.1:5602
```

20.9.4 Optional configuration

1. Register backup/snapshot repository for Elasticsearch

```
bash
curl -k -XPUT "https://127.0.0.1:9200/_snapshot/backup?pretty" -H 'Content-Type:
→application/json' -d'
{
  "type": "fs",
  "settings": {
    "location": "/var/lib/elasticsearch/backup/"
  }
}' -u logserver:logserver
```

2. Login using curl/kibana

```
bash
curl -k -XPOST 'https://127.0.0.1:5602/auth/login' -H 'mimeType: application/x-
→www-form-urlencoded' -d 'user=logserver&password=logserver' -c cookie.txt
curl -k -XGET 'https://127.0.0.1:5602' -b cookie.txt
```

20.10 Curator - Elasticsearch index management tool

Curator is a tool that allows you to perform index management tasks, such as:

- Close Indices
- Delete Indices
- Delete Snapshots
- Forcemerge segments
- Changing Index Settings

- Open Indices

- Reindex data

And other.

20.10.1 Curator installation

Curator is delivered with the client node installer.

20.10.2 Curator configuration

Create directory for configuration:

```
mkdir /etc/curator
```

Create directory for Curator logs file:

```
mkdir /var/log/curator
```

20.10.3 Running Curator

The curator executable is located in the directory:

```
/usr/share/kibana/curator/bin/curator
```

Curator requires two parameters:

- config - path to configuration file for Curator

- path to action file for Curator

Example running command:

```
/usr/share/kibana/curator/bin/curator --config /etc/curator/curator.conf /etc/curator/
→close_indices.yml
```

20.10.4 Sample configuration file

```
---
# Remember, leave a key empty if there is no value.  None will be a string,
# not a Python "NoneType"
client:
  hosts:
    - 127.0.0.1
  port: 9200
#  url_prefix:
#  use_ssl: False
#  certificate:
  client_cert:
  client_key:
  ssl_no_validate: False
  http_auth: $user:$passowrd
```

(continues on next page)

```
  timeout: 30
  master_only: True

logging:
  loglevel: INFO
  logfile: /var/log/curator/curator.log
  logformat: default
  blacklist: ['elasticsearch', 'urllib3']
```

20.10.5 Sample action file

- close indices

```
actions:
  1:
    action: close
    description: >-
      Close indices older than 30 days (based on index name), for logstash-
      prefixed indices.
    options:
      delete_aliases: False
      timeout_override:
      continue_if_exception: False
      disable_action: True
    filters:
    - filtertype: pattern
      kind: prefix
      value: logstash-
      exclude:
    - filtertype: age
      source: name
      direction: older
      timestring: '%Y.%m.%d'
      unit: days
      unit_count: 30
      exclude:
```

- delete indices

```
actions:
  1:
    action: delete_indices
    description: >-
      Delete indices older than 45 days (based on index name), for logstash-
      prefixed indices. Ignore the error if the filter does not result in an
      actionable list of indices (ignore_empty_list) and exit cleanly.
    options:
      ignore_empty_list: True
      timeout_override:
      continue_if_exception: False
      disable_action: True
    filters:
    - filtertype: pattern
      kind: prefix
      value: logstash-
```

```
      exclude:
    - filtertype: age
      source: name
      direction: older
      timestring: '%Y.%m.%d'
      unit: days
      unit_count: 45
      exclude:
```

- forcemerge segments

```
actions:
  1:
    action: forcemerge
    description: >-
      forceMerge logstash- prefixed indices older than 2 days (based on index
      creation_date) to 2 segments per shard.  Delay 120 seconds between each
      forceMerge operation to allow the cluster to quiesce.
      This action will ignore indices already forceMerged to the same or fewer
      number of segments per shard, so the 'forcemerged' filter is unneeded.
    options:
      max_num_segments: 2
      delay: 120
      timeout_override:
      continue_if_exception: False
      disable_action: True
    filters:
    - filtertype: pattern
      kind: prefix
      value: logstash-
      exclude:
    - filtertype: age
      source: creation_date
      direction: older
      unit: days
      unit_count: 2
      exclude:
```

- open indices

```
actions:
  1:
    action: open
    description: >-
      Open indices older than 30 days but younger than 60 days (based on index
      name), for logstash- prefixed indices.
    options:
      timeout_override:
      continue_if_exception: False
      disable_action: True
    filters:
    - filtertype: pattern
      kind: prefix
      value: logstash-
      exclude:
    - filtertype: age
      source: name
```

```
        direction: older
        timestring: '%Y.%m.%d'
        unit: days
        unit_count: 30
        exclude:
      - filtertype: age
        source: name
        direction: younger
        timestring: '%Y.%m.%d'
        unit: days
        unit_count: 60
        exclude:
```

- replica reduce

```
actions:
  1:
    action: replicas
    description: >-
      Reduce the replica count to 0 for logstash- prefixed indices older than
      10 days (based on index creation_date)
    options:
      count: 0
      wait_for_completion: False
      timeout_override:
      continue_if_exception: False
      disable_action: True
    filters:
    - filtertype: pattern
      kind: prefix
      value: logstash-
      exclude:
    - filtertype: age
      source: creation_date
      direction: older
      unit: days
      unit_count: 10
      exclude:
```

20.11 Cross-cluster Search

Cross-cluster search lets you run a single search request against one or more remote clusters. For example, you can use a cross-cluster search to filter and analyze log data stored on clusters in different data centers.

20.11.1 Configuration

1. Use `_cluster` API to add least one remote cluster:

```
curl -u user:password -X PUT "localhost:9200/_cluster/settings?pretty" -H 'Content-
→Type: application/json' -d'
{
  "persistent": {
    "cluster": {
```

```
      "remote": {
        "cluster_one": {
          "seeds": [
            "192.168.0.1:9300"
          ]
        },
        "cluster_two": {
          "seeds": [
            "192.168.0.2:9300"
          ]
        }
      }
    }
  }
}'
```

1. To search data in index `twitter` located on the `cluster_one` use following command:

```
curl -u user:password -X GET "localhost:9200/cluster_one:twitter/_search?pretty" -H
↪'Content-Type: application/json' -d'
{
  "query": {
    "match": {
      "user": "kimchy"
    }
  }
}'
```

1. To search data in index `twitter` located on multiple clusters, use following command:

```
curl -u user:password -X GET "localhost:9200/twitter,cluster_one:twitter,cluster_
↪two:twitter/_search?pretty" -H 'Content-Type: application/json' -d'
{
  "query": {
    "match": {
      "user": "kimchy"
    }
  }
}'
```

1. Configure index pattern in Kibana GUI to discover data from multiple clusters:

```
cluster_one:logstash-*,cluster_two:logstash-*
```

```
media/media/image133.png
```

20.11.2 Security

Cross-cluster search uses the Elasticsearch transport layer (default 9300/tcp port) to exchange data. To secure the transmission, encryption must be enabled for the transport layer.

Configuration is in the `/etc/elasticsearch/elastisearch.yml` file:

```
# Transport layer encryption
logserverguard.ssl.transport.enabled: true
logserverguard.ssl.transport.pemcert_filepath: "/etc/elasticsearch/ssl/certificate.crt
↪"
logserverguard.ssl.transport.pemkey_filepath: "/etc/elasticsearch/ssl/certificate.key"
logserverguard.ssl.transport.pemkey_password: ""
logserverguard.ssl.transport.pemtrustedcas_filepath: "/etc/elasticsearch/ssl/rootCA.
↪crt"

logserverguard.ssl.transport.enforce_hostname_verification: false
logserverguard.ssl.transport.resolve_hostname: false
```

Encryption must be enabled on each cluster.

20.12 Sync/Copy

The Sync/Copy module allows you to synchronize or copy data between two Elasticsearch clusters. You can copy or synchronize selected indexes or indicate index pattern.

20.12.1 Configuration

Before starting Sync/Copy, complete the source and target cluster data in the `Profile` and `Create profile`tab:

- Protocol - http or https;
- Host - IP address ingest node;
- Port - communication port (default 9200);
- Username - username that has permission to get data and save data to the cluster;
- Password - password of the above user
- Cluster name

media/media/image134.png

You can view or delete the profile in the `Profile List` tab.

20.12.2 Synchronize data

To perform data synchronization, follow the instructions:

- go to the `Sync` tab;
- select `Source Profile`
- select `Destination Profile`
- enter the index pattern name in `Index pattern to sync`

- or use switch `Toggle to select between Index pattern or name` and enter indices name.
- to create synchronization task, press `Submit` button

media/media/image135.png

20.12.3 Copy data

To perform data copy, follow the instructions:

- go to the `Copy` tab;
- select `Source Profile`
- select `Destination Profile`
- enter the index pattern name in `Index pattern to sync`
- or use switch `Toggle to select between Index pattern or name` and enter indices name.
- to start copying data press the `Submit` button

Logged in as : logserver

Sync	Copy	Jobs	Profile

Source Profile

192.168.3.221	⌄

Destination Profile

elasticsearch	⌄

☑◯ Toggle to select between Index pattern or name
Index pattern to copy

logstash-*

Indices to copy

⌄

Submit

20.12.4 Running Sync/Copy

Prepared Copy/Sync tasks can be run on demand or according to a set schedule. To do this, go to the `Jobs` tab. With each task you will find the `Action` button that allows:

- running the task;
- scheduling task in Cron format;
- deleting task;
- download task logs.

20.13 XLSX Import

The XLSX Import module allow to import your `xlsx` and `csv` file to indices.

20.13.1 Importing steps

1. Go to XLSX Import module and select your file and sheet:

After the data has been successfully loaded, you will see a preview of your data at the bottom of the window.

Press `Next` button.

1. In the next step, enter the index name in the `Index name` field, you can also change the pattern for the document ID and select the columns that the import will skip.

Index name

task_logs

Name the elasticsearch index that will be created. If the index is
already existing, documents will be added or updated according to the
chosen docID

Custom docID

line{_line}-{_uid}   example rendering

line1337-ePqwGNw3dsJU

Import will provide a unique document identifier linked to the line
number of the imported file. You can customize this doc ID using
special reserved variables : {_uid} for an auto-generated identifier,
{_importedLine} for the current line number, or {<column-name>} to
access a value of the imported line.

Removing columns

∨
Date
Duration
Task

Europe/Berlin   ∨

Excel does not manage timezone within date format cells. Define your
file content timezone to index its date fields in a correct way.

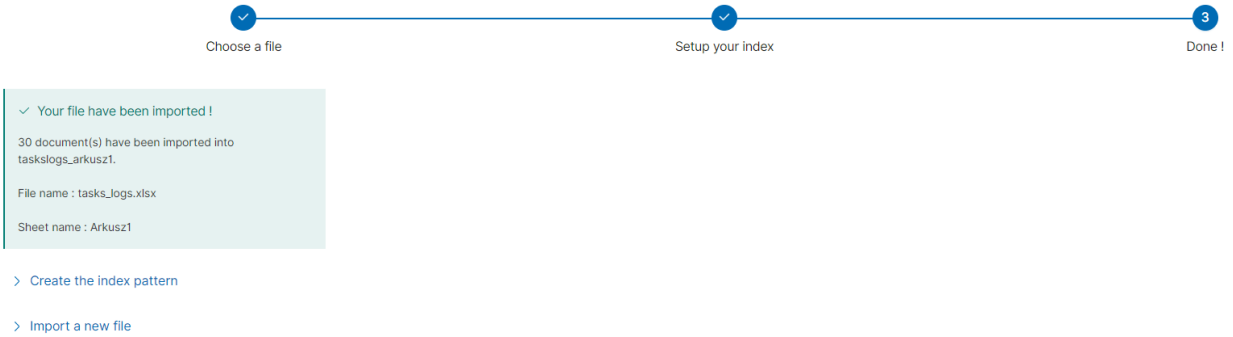◯ ✕ Configure your own mapping ⓘ

◯ ✕ Add ingest pipeline ids ⓘ

‹ back   ⬇ Import

1. Select the `Configure your own mapping` for every field. You can choose the type
   and apply more options with the advanced JSON. The list of parameters can be found here,
   https://www.elastic.co/guide/en/elasticsearch/reference/7.x/mapping-params.html

2. After the import configuration is complete, select the `Import` button to start the import process.

3. After the import process is completed, a summary will be displayed. Now you can create a new index pattern to view your data in the Discovery module.

XLSX Import

Choose a file   Setup your index   Done !

✓ Your file have been imported !

30 document(s) have been imported into taskslogs_arkusz1.

File name : tasks_logs.xlsx

Sheet name : Arkusz1

> Create the index pattern

> Import a new file

Troubleshooting

21.1 Recovery default base indexes

Only applies to versions 6.1.5 and older. From version 6.1.6 and later, default indexes are created automatically

If you lost or damage following index:

```
|Index name            | Index ID                |
|----------------|----------------------|
| .security      |Pfq6nNXOSSmGhqd2fcxFNg |
| .taskmanagement|E2Pwp4xxTkSc0gDhsE-vvQ |
| alert_status   |fkqks4J1QnuqiqYmOFLpsQ |
| audit          |cSQkDUdiSACo9WlTpc1zrw |
| alert_error    |9jGh2ZNDRumU0NsB3jtDhA |
| alert_past     |1UyTN1CPTpqm8eDgG9AYnw |
| .trustedhost   |AKKfcpsATj6M4B_4VD5vIA |
| .kibana        |cmN5W7ovQpW5kfaQ1xqf2g |
| .scheduler_job |9G6EEX9CSEWYfoekNcOEMQ |
| .authconfig    |2M01Phg2T-q-rEb2rbfoVg |
| .auth          |ypPGuDrFRu-_ep-iYkgepQ |
| .reportscheduler|mGroDs-bQyaucfY3-smDpg |
| .authuser      |zXotLpfeRnuzOYkTJpsTaw |
| alert_silence  |ARTo7ZwdRL67Khw_HAIkmw |
| .elastfilter   |TtpZrPnrRGWQlWGkTOETzw |
| alert          |RE6EM4FfR2WTn-JsZIvm5Q |
| .alertrules    |SzV22qrORHyY9E4kGPQOtg |
```

You may to recover it from default installation folder with following steps:

1. Stop Logstash instances which load data into cluster

```
systemctl stop logstash
```

2. Disable shard allocation

```
PUT _cluster/settings
{
  "persistent": {
    "cluster.routing.allocation.enable": "none"
  }
}
```

3. Stop indexing and perform a synced flush

```
POST _flush/synced
```

4. Shutdown all nodes:

```
systemctl stop elasticsearch.service
```

5. Copy appropriate index folder from installation folder to Elasticsearch cluster data node folder (example of .auth folder)

```
cp -rf ypPGuDrFRu-_ep-iYkgepQ /var/lib/elasticsearch/nodes/0/indices/
```

6. Set appropriate permission

```
chown -R elasticsearch:elasticsearch /var/lib/elasticsearch/
```

7. Start all Elasticsearch instance

```
systemctl start elasticsearch
```

8. Wait for yellow state of Elasticsearch cluster and then enable shard allocation

```
PUT _cluster/settings
{
  "persistent": {
    "cluster.routing.allocation.enable": "all"
  }
}
```

9. Wait for green state of Elasticsearch cluster and then start the Logstash instances

```
systemctl start logstash
```

21.2 Too many open files

If you have a problem with too many open files by the Elasticsearch process, modify the values in the following configuration files:

- /etc/sysconfig/elasticsearch
- /etc/security/limits.d/30-elasticsearch.conf
- /usr/lib/systemd/system/elasticsearch.service

Check these three files for:

- LimitNOFILE=65536
- elasticsearch nofile 65537

- MAX_OPEN_FILES=65537

Changes to service file require:

```
systemctl daemon-reload
```

And changes to limits.d require:

```
sysctl -p /etc/sysctl.d/90-elasticsearch.conf
```

21.3 The Kibana status code 500

If the login page is displayed in Kibana, but after the attempt to login, the browser displays "error: 500", and the logs will show entries:

```
Error: Failed to encode cookie (sid-auth) value: Password string too short (min 32
→characters required).
```

Generate a new server.ironsecret with the following command:

```
echo "server.ironsecret: \"$(</dev/urandom tr -dc _A-Z-a-z-0-9 | head -c32)\"" >> /
→etc/kibana/kibana.yml
```

21.4 Diagnostic tool

ITRS Log-Analytics includes a diagnostic tool that helps solve your problem by collecting system data necessary for problem analysis by the support team.

21.4.1 Location

The diagnostic tool is located in the installation directory: `utils/diagnostic-tool.sh`

21.4.2 Gathering information

Diagnostic tool collect the following information:

- configuration files for Kibana, Elasticsearch, Alert
- logs file for Kibana, Alert, Cerebro, Elasticsearch
- Cluster information from Elasticsearch API

When the diagnostic tool collects data, the credentials are removed from the content of the files.

21.4.3 Running the diagnostic tool

To run the diagnostic tool, you must provide three parameters: - user assigned admin role, default 'logserver' - user password; - URL of cluster API, default: `http://localhost:9200`

Example of a command:

```
./diagnostic-tool.sh $user $password http://localhost:9200
```

The diagnostic tool saves the results to `.tar` file located in the user's home directory.

Upgrades

You can check the current version using the API command:

```
curl -u $USER:$PASSWORD -X GET http://localhost:9200/license
```

22.1 Upgrade from version 7.0.1

22.1.1 General note

- Update the `kibana` role to include index-pattern `.kibana*`

- Update the `alert` role to include index-pattern `.alertrules*` and `alert_status*`

- Install `python36` which is required for the Alerting engine

```
yum install python3 on client-node
```

- AD users should move their saved objects from the `adrole`.

- Indicators of compromise (IOCs auto-update) require access to the software provider's servers.

- GeoIP Databases (auto-update) require access to the software provider's servers.
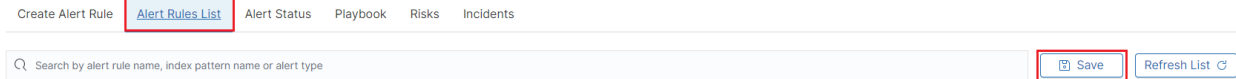
22.1.2 Upgrade steps

- Stop services

```
systemctl stop elasticsearch alert kibana
```

- Upgrade client-node (includes alert engine)

```
yum update ./itrs-log-analytics-client-node-7.0.2-1.el7.x86_64.rpm
```

- Login in the GUI ITRS Log Analytics and go to the `Alert List` on the `Alerts` tab and click `SAVE` button

Create Alert Rule	Alert Rules List	Alert Status	Playbook	Risks	Incidents

🔍 Search by alert rule name, index pattern name or alert type	🖫 Save	Refresh List ↻

- Start `alert` and `kibana` service

```
systemctl start alert kibana
```

- Upgrade data-node

```
yum update ./itrs-log-analytics-data-node-7.0.2-1.el7.x86_64.rpm
```

- Start services

```
systemctl start elasticsearch
```

Extra note

If the Elasticsearch service has been started on the client-node, then it is necessary to update the **client.rpm** and **data.rpm** packages on the client node.

After update, you need to edit:

```
/etc/elasticsearch/elasticsearch.yml
```

and change:

```
node.data: false
```

Additionally, check the file:

```
elasticsearch.yml.rpmnew
```

and complete the configuration in `elasticsearch.yml` with additional lines.

22.2 Changing OpenJDK version

22.2.1 Logstash

OpenJDK 11 is supported by Logstash from version 6.8 so if you have an older version of Logstash you must update it.

To update Logstash, follow the steps below:

1. Back up the following files

- /etc/logstash/logstash.yml
- /etc/logstash/piplines.yml
- /etc/logstash/conf.d

1. Use the command to check custom Logstash plugins:

```
/usr/share/bin/logstash-plugin list --verbose
```

and note the result

2. Install a newer version of Logstash according to the instructions:

   https://www.elastic.co/guide/en/logstash/6.8/upgrading-logstash.html

   or

   https://www.elastic.co/guide/en/logstash/current/upgrading-logstash.html

3. Verify installed plugins:

   ```
   /usr/share/bin/logstash-plugin list --verbose
   ```

4. Install the missing plugins if necessary:

   ```
   /usr/share/bin/logstash-plugin install plugin_name
   ```

5. Run Logstash using the command:

   ```
   systemctl start logstash
   ```

22.2.2 Elasticsearch

ITRS Log Analytics can use OpenJDK version 10 or later. If you want to use OpenJSK version 10 or later, configure the Elasticsearch service as follows:

1. After installing OpenJDK, select the correct version that Elasticsearch will use:

   ```
   alternative --config java
   ```

2. Open the /etc/elasticsearch/jvm.options file in a text editor:

   ```
   vi /etc/elasticsearch/jvm.options
   ```

3. Disable the OpenJDK version 8 section:

   ```
   ## JDK 8 GC logging

   #8:-XX:+PrintGCDetails
   #8:-XX:+PrintGCDateStamps
   #8:-XX:+PrintTenuringDistribution
   #8:-XX:+PrintGCApplicationStoppedTime
   #8:-Xloggc:/var/log/elasticsearch/gc.log
   #8:-XX:+UseGCLogFileRotation
   #8:-XX:NumberOfGCLogFiles=32
   #8:-XX:GCLogFileSize=64m
   ```

4. Enable the OpenJDK version 11 section

   ```
   ## G1GC Configuration
   # NOTE: G1GC is only supported on JDK version 10 or later.
   # To use G1GC uncomment the lines below.
   10-:-XX:-UseConcMarkSweepGC
   10-:-XX:-UseCMSInitiatingOccupancyOnly
   10-:-XX:+UseG1GC
   10-:-XX:InitiatingHeapOccupancyPercent=75
   ```

5. Restart the Elasticsearch service

```
systemctl restart elasticsearch
```

Agents module

The Agents module is used for the central management of agents used in Energy Logserver such as Filebeat, Winlogbeat, Packetbeat, Metricbeat.# Agent installation # All necessary components can be found in the installation folder *${installation_folder}/utils/agents_bin*.

23.1 Component modules

The software consists of two modules:

- Plugin Agents - installation just like any standard Kibana plugin. Before you run the module for the first time, you must add the mapping for the .agents index with the `create_temlate.sh` script

- MasterAgent software - installed on host with agent (like beats);

23.2 Table of configuration parameter for Agent software

```
     |Parameter             |Work type           |Required |Defult value
↪ |Description                  |
     |---------------------|--------------------|---------|--------------------
↪-|----------------------------------------|
     |port                 |Agent               |No       |40000
↪ |The port on which |the agent is listening  |
     |host                 |Agent               |No       |Read from system
↪ |The address on which the agent is listening |
     |hostname             |Agent               |No       |Read from system
↪ |Host name (hostname)                  |
     |autoregister         |Agent               |No       |24
↪ |How often the agent's self-registration should take place. Time in hours |
     |metricbeat_path      |Agent               |No       |./
↪ |Catalog for meatricbeat                  |
     |filebeat_path        |Agent               |No       |./
↪ |Directory for filebeat                  |
```

(continues on next page)

<div style="text-align: right">(continued from previous page)</div>

```
        |winlogbeat_path     |Agent               |No      |./
→ |Catalog for winlogbeat                      |
        |packetbeat_path     |Agent               |No      |./
→ |Catalog for packetbeat                      |
        |custom_list         |Agent               |No      |Not defiend
→ |List of files and directories to scan. If a directory is specified, files with
→the yml extension are registered with it. The file / directory separator is the
→character ";" |
        |createfile_folder   |Agent               |No      |Not defiend
→ |List of directories where files can be created. The catalogs are separated by the
→symbol ";". These directories are not scanned for file registration.
        |logstash            |Agent               |No      |https://
→localhost:8080 |Logstash address for agents                  |
        |https_keystore      |Agent and Masteragent |No      |./lig.keystore
→ |Path to the SSL certificate file.          |
        |https_keystore_pass |Agent and Masteragent |No      |admin
→ |The password for the certificate file      |
        |connection_timeout  |Agent and Masteragent |No      |5
→ |Timeout for https calls given in seconds.    |
        |connection_reconnect |Agent and Masteragent |No      |5
→ |Time in seconds that the agent should try to connect to the Logstash if error
→occur |
```

23.3 Installing agent software

The Agent's software requires the correct installation of a Java Runtime Environment. The software has been tested on Oracle Java 8. It is recommended to run the Agent as a service in a given operating system.

1. Generating the certificates - EDIT DOMAIN, DOMAIN_IP - use this scripts:

   - create CA certificate and key:

```bash
#!/bin/bash
DOMAIN="localhost"
DOMAIN_IP="192.168.0.1"
COUNTRYNAME="PL"
STATE="Poland"
COMPANY="ACME"

openssl genrsa -out rootCA.key 4096

echo -e "${COUNTRYNAME}\n${STATE}\n\n${COMPANY}\n\n\n\n" | openssl    req -
→x509 -new -nodes -key rootCA.key -sha256 -days 3650 -out rootCA.crt
```

   - create certificate and key for you domain:

```bash
#!/bin/bash
DOMAIN="localhost"
DOMAIN_IP="192.168.0.1"
COUNTRYNAME="PL"
STATE="Poland"
COMPANY="ACME"
openssl genrsa -out ${DOMAIN}.pre 2048
openssl pkcs8 -topk8 -inform pem -in ${DOMAIN}.pre -outform pem -out ${DOMAIN}.
 →key -nocrypt
```

<div style="text-align: right">(continues on next page)</div>

```
openssl req -new -sha256 -key ${DOMAIN}.key -subj "/C=${COUNTRYNAME}/ST=${STATE}/
↪O=${COMPANY}/CN=${DOMAIN}" -reqexts SAN -config <(cat /etc/pki/tls/openssl.cnf
↪<(printf "[SAN]\nsubjectAltName=DNS:${DOMAIN},IP:${DOMAIN_IP}")) -out ${DOMAIN}.
↪csr

openssl x509 -req -in ${DOMAIN}.csr -CA rootCA.crt -CAkey rootCA.key -
↪CAcreateserial -out ${DOMAIN}.crt -sha256 -extfile <(printf "[req]\ndefault_
↪bits=2048\ndistinguished_name=req_distinguished_name\nreq_extensions=req_
↪ext\n[req_distinguished_name]\ncountryName=${COUNTRYNAME}\nstateOrProvinceName=$
↪{STATE}\norganizationName=${COMPANY}\ncommonName=${DOMAIN}\n[req_
↪ext]\nsubjectAltName=@alt_names\n[alt_names]\nDNS.1=${DOMAIN}\nIP=${DOMAIN_IP}\n
↪") -days 3650 -extensions req_ext
```

- to verify certificate use following command:

```
openssl x509 -in ${DOMAIN}.crt -text -noout
```

- creating Java keystore, you will be asked for the password for the certificate key and whether the certificate should be trusted - enter "yes"

```
#!/bin/bash
DOMAIN="localhost"
DOMAIN_IP="192.168.0.1"
COUNTRYNAME="PL"
STATE="Poland"
COMPANY="ACME"
 keytool -import -file rootCA.crt -alias root -keystore root.jks -storetype jks
openssl pkcs12 -export -in ${DOMAIN}.crt -inkey ${DOMAIN}.pre -out    node_name.
↪p12 -name "${DOMAIN}" -certfile rootCA.crt
```

2. Linux host configuration

- To install the MasterAgent on Linux RH / Centos, the net-tools package must be installed:

```
yum install net-tools
```

- Add an exception to the firewall to listen on TCP 8080 and 8081:

```
firewall-cmd --permanent --zone public --add-port 8080/tcp
firewall-cmd --permanent --zone public --add-port 8081/tcp
```

- Logstash - Configuration

```
/bin/cp -rf ./logstash/agents_template.json /etc/logstash/templates.d/
mkdir /etc/logstash/conf.d/masteragent
/bin/cp -rf ./logstash/*.conf /etc/logstash/conf.d/masteragent/

 /etc/logstash/pipelines.yml:
- pipeline.id: masteragent
 path.config: "/etc/logstash/conf.d/masteragent/*.conf"

mkdir /etc/logstash/conf.d/masteragent/ssl
/bin/cp -rf ./certificates/localhost.key /etc/logstash/conf.d/masteragent/ssl/
/bin/cp -rf ./certificates/localhost.crt /etc/logstash/conf.d/masteragent/ssl/
/bin/cp -rf ./certificates/rootCA.crt /etc/logstash/conf.d/masteragent/ssl/
chown -R logstash:logstash /etc/logstash
```

- Masterbeat - Installation

```
/bin/cp -rf ./agents/linux /opt/agents
/bin/cp -rf ./agents/linux/agents/linux/MasterBeatAgent.conf /opt/agents/agent.conf
/bin/cp -rf ./certificates/node_name.p12 /opt/agents/
/bin/cp -rf ./certificates/root.jks /opt/agents/
chown -R kibana:kibana /opt/agents
```

- Linux Agent - Installation

```
/bin/cp -rf ./agents/linux/masteragent /opt/masteragent
/bin/cp -rf ./certificates/node_name.p12 /opt/masteragent
/bin/cp -rf ./certificates/root.jks /opt/masteragent
/bin/cp -rf ./agents/linux/masteragent/masteragent.service
/usr/lib/systemd/system/masteragent.service
systemctl daemon-reload
systemctl enable masteragent
systemctl start masteragent
```

- Download `MasterBeatAgent.jar` and `agent.conf` files to any desired location;

- Upload a file with certificates generated by the `keytool` tool to any desired location;

- Update entries in the `agent.conf` file (the path to the key file, paths to files and directories to be managed, the Logstash address, etc.);

```
The agent should always be run with an indication of the working directory in
↪which the `agent.conf` file is located;
```

- The Agent is started by the `java -jar MasterBeatAgent.jar` command.

- Configuration of the `/etc/systemd/system/masteragent.service` file:

```
        [Unit]
            Description=Manage MasterAgent service
            Wants=network-online.target
            After=network-online.target

            [Service]
            WorkingDirectory=/opt/agent
            ExecStart=/bin/java -jar MasterBeatAgent.jar
            User=root
            Type=simple
            Restart=on-failure
            RestartSec=10

            [Install]
            WantedBy=multi-user.target
```

- After creating the file, run the following commands:

```
  systemctl daemon-reload
  systemctl enable  masteragent
  systemctl start masteragent
```

1. Windows host configuration

- Download the latest version of MasterAgnet, which includes:

    - Agents.jar;

    - agents.exe;

- – agent.conf;

- – agents.xml;

- – lig.keystore;

- Add an exception to the firewall to listen on TCP port 8081;

- Add an exception to the firewall to allow connection on TCP port 8080 with remote hosts;

- Copy Master Agent files to installation directory: "C:\Program Files\MasterAgent"

- To install the service, start the PowerShell console as an administrator and execute the following commands:

```
New-Service -name masteragent -displayName masteragent -binaryPathName
→"C:\Program Files\MasterAgent\agents.exe"
```

- Check status of the services

```
cd C:\Program Files\MasterAgent
agents.exe status
```

23.4 TLS configuration

The default agent uses TLS 1.2 for communication. In addition, you can disable the agent's ability to use weak protocols and change other cryptographic options, such as the length of the Diffie-Hellman key.

- Create a configuration file `agent.security`:

```
vi /opt/agent/agent.security
```

- Add the necessary configuration, for example:

```
jdk.tls.disabledAlgorithms=SSLv2Hello, SSLv3, TLSv1, TLSv1.1, RC4, DES,
→MD5withRSA, DH keySize < 2048, \
    EC keySize < 224, 3DES_EDE_CBC, anon, NULL
```

- Add a new configuration to the service unit:

```
systemctl edit --full masteragent.service
```

- Add the following line:

```
[Unit]
Description=Manage MasterAgent service
Wants=network-online.target
After=network-online.target

[Service]
WorkingDirectory=/opt/agent
- ExecStart=/bin/java -jar MasterBeatAgent.jar
+ ExecStart=/bin/java -Djava.security.properties=/opt/agent/agent.security -jar
→MasterBeatAgent.jar
User=root
Type=simple
Restart=on-failure
RestartSec=10
```

```
[Install]
WantedBy=multi-user.target
```

- Reload daemon and restart service

```
systemctl daemon-reload
systemctl restart masteragent.service
```

23.5 The agent management

The GUI console is used to manage agents. In the **Agetns** tab, you can find a list of connected agents. There are typical information about agents such as:

- Host name;

- OS name;

- IP Address;

- TCP port;

- Last revision;

Additionally, for each connected agent, you can find action buttons such as:

- Drop - to remove the agent configuration from the GUI;

- Create - to create new configuration files;

- Show - it is used to display the list of created configuration files;

23.5.1 Creating a new configuration file

To add a new configuration file press the **Create** button, add a new file **name**, add a new **path** where the file should be saved and the context of the new configuration file. The new file will be saved with the extension * .yml.

23.5.2 Editing configuration file

To display a list of configuration files available for a given host, press the Show button.

A list of configuration files will be displayed, and the following options for each of them:

- Show - displays the contents of the file;
- Edit - edit the contents of the file;
- Delete - deletes the file.

To edit the file, select the Edit button, then enter the changes in the content window, after finishing select the Submit button.

Monitoring

24.1 Skimmer

ITRS Log Analytics uses a monitoring module called Skimmer to monitor the performance of its hosts. Metrics and conditions of services are retrieved using the API.

The services that are supported are:

- Elasticsearch;

- Logstash;

- Kibana;

- Metricbeat;

- Pacemaker;

- Zabbix;

- Zookeeper;

- Kafka;

- Httpbeat;

- Elastalert;

- Filebeat

and other.

24.2 Skimmer Installation

The RPM package skimmer-x86_64.rpm is delivered with the system installer in the "utils" directory:

```
        cd $install_direcorty/utils
        yum install skimmer-x86_64.rpm -y
```

24.3 Skimmer service configuration

The Skimmer configuration is located in the `/usr/share/skimmer/skimmer.conf` file.

```
# index name in elasticsearch
index_name = skimmer
index_daily = true
# type in elasticsearch index
index_type = _doc
# user and password to elasticsearch api
elasticsearch_auth = logserver:logserver
# available outputs
elasticsearch_address = 127.0.0.1:9200
# logstash_address = 127.0.0.1:6110
# retrieve from api
elasticsearch_api = 127.0.0.1:9200
# logstash_api = 127.0.0.1:9600
# path to log file
log_file = /tmp/skimmer.log
# daemonize
daemonize = true
# comma separated OS statistics selected from the list [zombie,vm,fs,swap,net,cpu]
os_stats = zombie,vm,fs,swap,net,cpu
# comma separated process names to print their pid
processes = /usr/sbin/sshd,/usr/sbin/rsyslogd
# comma separated systemd services to print their status
systemd_services = elasticsearch,logstash,kibana
# comma separated port numbers to print if address is in use
port_numbers = 9200,9300,9600,5601
# path to directory containing files needed to be csv validated
csv_path = /tmp/csv_dir
```

After the changes in the configuration file, restart the service.

```
systemctl restart skimmer
```

24.3.1 Skimmer GUI configuration

To view the collected data by the skimmer in the GUI, you need to add an index pattern.

Go to the "**Management**" -> "**Index Patterns**" tab and press the "**Create Index Pattern**" button. In the "**Index Name**" field, enter the formula `skimmer- *`, and select the "**Next step**" button. In the "**Time Filter**" field, select `@timestamp` and then press "**Create index pattern**"

In the "**Discovery**" tab, select the `skimmer- *` index from the list of indexes. A list of collected documents with statistics and statuses will be displayed.

Kafka

25.1 Enabling encryption for Apache Kafka clients

Kafka allows you to distribute the load between nodes receiving data and encrypts communication.

Architecture example:
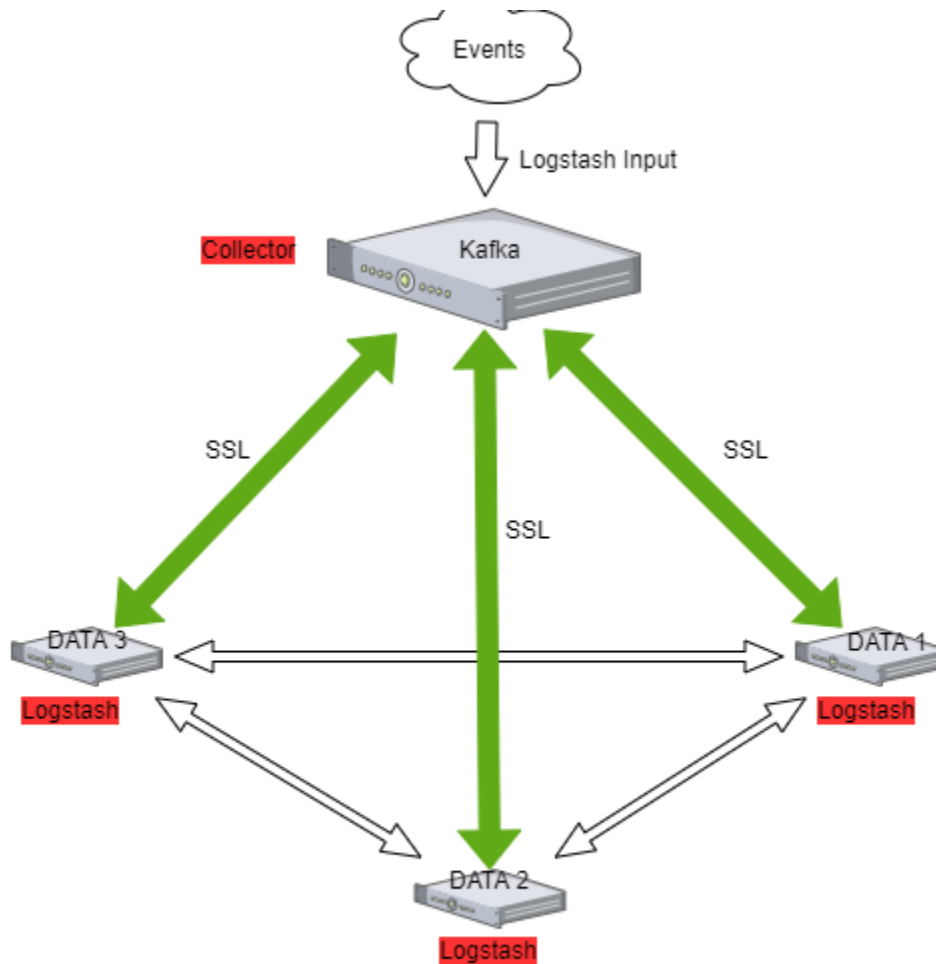
25.1.1 The Kafka installation

Documentation during creation.

25.1.2 Enabling encryption in Kafka

Generate SSL key and certificate for each Kafka broker

```
keytool -keystore server.keystore.jks -alias localhost -validity {validity} -genkey -
→keyalg RSA
```

Configuring Host Name In Certificates

```
keytool -keystore server.keystore.jks -alias localhost -validity {validity} -genkey -
→keyalg RSA -ext SAN=DNS:{FQDN}
```

Verify content of the generated certificate:

```
keytool -list -v -keystore server.keystore.jks
```

Creating your own CA

```
openssl req -new -x509 -keyout ca-key -out ca-cert -days 365
keytool -keystore client.truststore.jks -alias CARoot -import -file ca-cert
keytool -keystore server.truststore.jks -alias CARoot -import -file ca-cert
```

Signing the certificate

```
keytool -keystore server.keystore.jks -alias localhost -certreq -file cert-file
openssl x509 -req -CA ca-cert -CAkey ca-key -in cert-file -out cert-signed -days
↪{validity} -CAcreateserial -passin pass:{ca-password}
```

Import both the certificate of the CA and the signed certificate into the keystore

```
keytool -keystore server.keystore.jks -alias CARoot -import -file ca-cert
keytool -keystore server.keystore.jks -alias localhost -import -file cert-signed
```

If you have trusted certificates, you must import them into the JKS keystore as follows:

Create a keystore:

```
keytool -keystore client.keystore.jks -alias localhost -validity 365 -keyalg RSA -
↪genkey
```

Combine the certificate and key file into a certificate in p12 format:

```
openssl pkcs12 -export -in cert_name.crt -inkey key_name.key -out cert_name.p12 -name␣
↪localhost -CAfile ca.crt -caname root
```

Import the CA certificate into a truststore:

```
keytool -keystore client.truststore.jks -alias CARoot -import -file ca-cert
```

Import the CA certificate into a keystore:

```
keytool -keystore client.keystore.jks -alias CARoot -import -file ca-cert
```

Import the p12 certificate into a keystore:

```
keytool -importkeystore -deststorepass MY-KEYSTORE-PASS -destkeystore client.keystore.
↪jks -srckeystore cert_name.p12 -srcstoretype PKCS12
```

25.1.3 Configuring Kafka Brokers

In `/etc/kafka/server.properties` file set the following options:

```
listeners=PLAINTEXT://host.name:port,SSL://host.name:port

ssl.keystore.location=/var/private/ssl/server.keystore.jks
ssl.keystore.password=test1234
ssl.key.password=test1234
ssl.truststore.location=/var/private/ssl/server.truststore.jks
ssl.truststore.password=test1234
```

and restart the Kafka service

```
systemctl restart kafka
```

25.1.4 Configuring Kafka Clients

Logstash

Configure the output section in Logstash based on the following example:

```
output {
  kafka {
    bootstrap_servers => "host.name:port"
    security_protocol => "SSL"
    ssl_truststore_type => "JKS"
    ssl_truststore_location => "/var/private/ssl/client.truststore.jks"
    ssl_truststore_password => "test1234"
    client_id => "host.name"
    topic_id => "Topic-1"
    codec => json
  }
}
```

Configure the input section in Logstash based on the following example:

```
input {
  kafka {
    bootstrap_servers => "host.name:port"
    security_protocol => "SSL"
    ssl_truststore_type => "JKS"
    ssl_truststore_location => "/var/private/ssl/client.truststore.jks"
    ssl_truststore_password => "test1234"
    consumer_threads => 4
    topics => [ "Topic-1" ]
    codec => json
    tags => ["kafka"]
  }
}
```

25.2 Log retention for Kafka topic

The Kafka durably persists all published records—whether or not they have been consumed—using a configurable retention period. For example, if the retention policy is set to two days, then for the two days after a record is published, it is available for consumption, after which it will be discarded to free up space. Kafka's performance is effectively constant with respect to data size so storing data for a long time is not a problem.

CHANGELOG

26.1 Version 7.0.2

26.1.1 New Features

- Manual incident - creating manual incidents from the Discovery section

- New kibana plugin - Sync/Copy between clusters

- Alert: Analyze historical data with defined alert

- Indicators of compromise (IoC) - providing blacklists based on Malware Information Sharing Platform (MISP)

- Automatic update of MaxMind GeoIP Databases [asn, city, country]

- Extended LDAP support

- Cross cluster search

- Diagnostic script to collect information about the environment, log files, configuration files - utils/diagnostic-tool.sh

- New beat: op5beat - dedicated data shipper from op5 Monitor

26.1.2 Improvements

- Added `_license` API for elasticsearch (it replaces `license` path which is now deprecated and will stop working in future releases)

- `_license` API now shows **expiration_date** and **days_left**

- Visual indicator on **Config** tab for expiring license (for 30 days and less)

- Creating a new user now requires reentering the passoword

- Complexity check for password fields

- Incidents can be supplemented with notes

- Alert Spike: more detailed description of usage

- ElasticDump added to base installation - /usr/share/kibana/elasticdump

- Alert plugin updated - frontend

- Reimplemented session timeout for user activity

- Skimmer: new metrics and dashboard for Cluster Monitoring

- Wazuh config/keys added to small_backup.sh script

- Logrotate definitions for Logtrail logfiles

- Incidents can be sorted by Risk value

- UTF-8 support for credentials

- Wazuh: wrong document_type and timestamp field

26.1.3 BugFixes

- Audit: Missing Audit entry for succesfull **SSO** login

- Report: "stderr maxBuffer length exceeded" - export to csv

- Report: "Too many scroll contexts" - export to csv

- Intelligence: incorrect work in updated environments

- Agents: fixed wrong document type

- Kibana: "Add Data to Kibana" from Home Page

- Incidents: the preview button uses the wrong index-pattern

- Audit: Missing information about login errors of ad/ldap users

- Netflow: fix for netflow v9

- MasterAgent: none/certificade verification mode should work as intended

- Incorrect CSS injections for dark theme

- The role could not be removed in specific scenarios

26.2 Version 7.0.1

- init

- migrated features from branch 6 [latest:6.1.8]

- XLSX import [kibana]

- curator added to /usr/share/kibana/curator

- node_modules updated! [kibana]

- elasticsearch upgraded to 7.3.2

- kibana upgraded to 7.3.2

- dedicated icons for all kibana modules

- eui as default framework for login,raports

- bugfix: alerts type description fix